Starter Kit for Arduino

Contents

Arduino IDE (Integrated Development Environment)

Introduction

The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

****

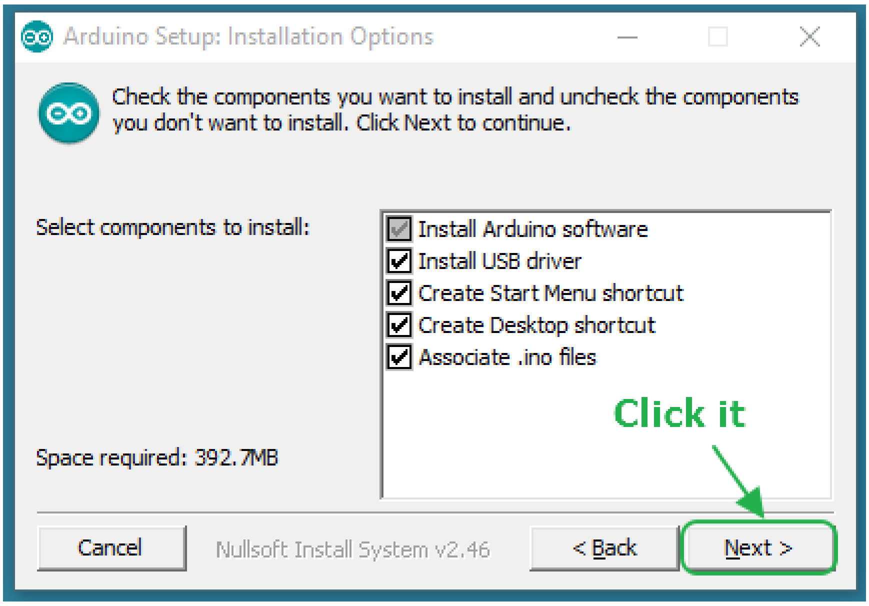
* About Elecrow:
* We are a leading manufacturer of electronic components for Arduino and Raspberry Pi.
* We have a professional engineering team dedicated to providing tutorials and support to help you get started.
* If you have any technical questions or suggestions, please feel free to contact our support staff via email at keen@elecrow.com
* We truly hope you enjoy the product, for more great products please visit our company website: https://www.elecrow.com
or Amazon store: www.amazon.com/shops/elecrow

Operation demo

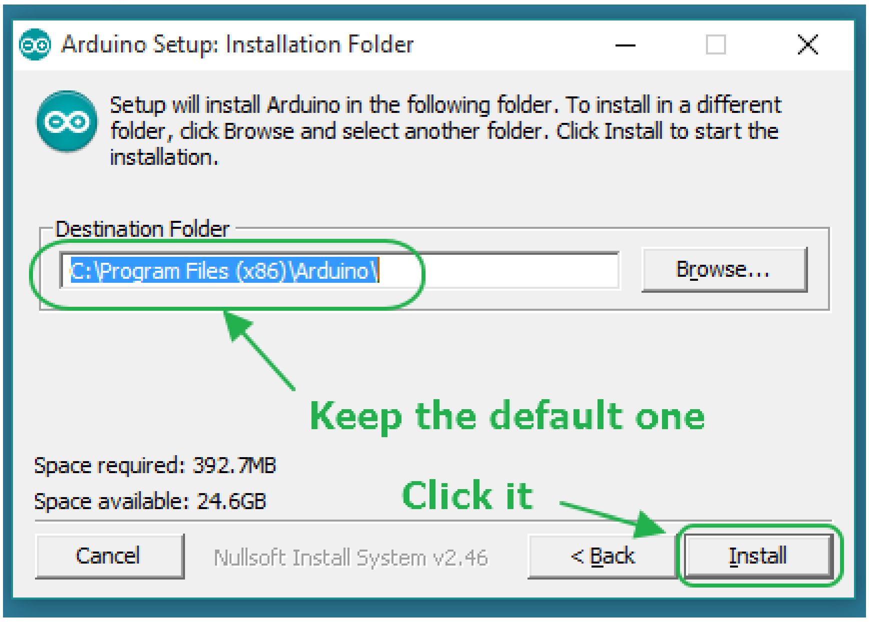
Step 1: Install the Arduino Software (IDE)

Download the latest version from this page: http://arduino.cc/en/Main/Software

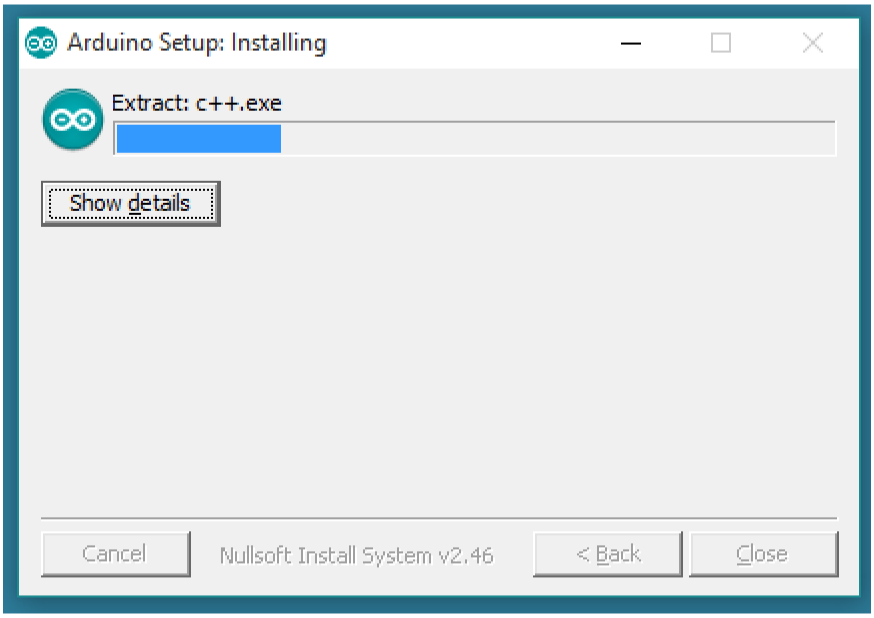
Next, proceed with the installation and please allow the driver installation process.



Choose the components to install and click “next” button.



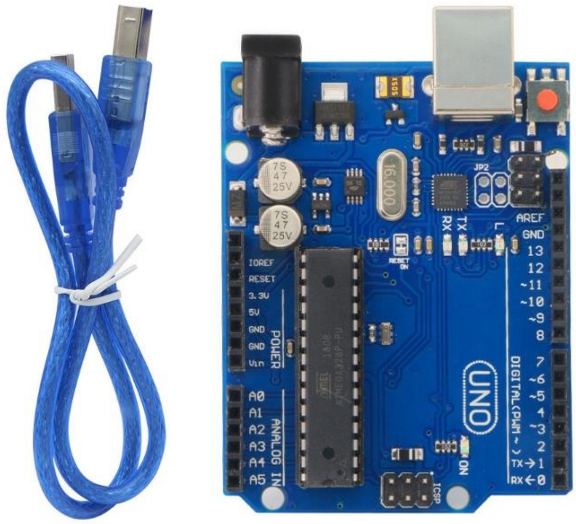
Choose the installation directory.



The process will extract and install all the required files to execute properly the Arduino Software (IDE)

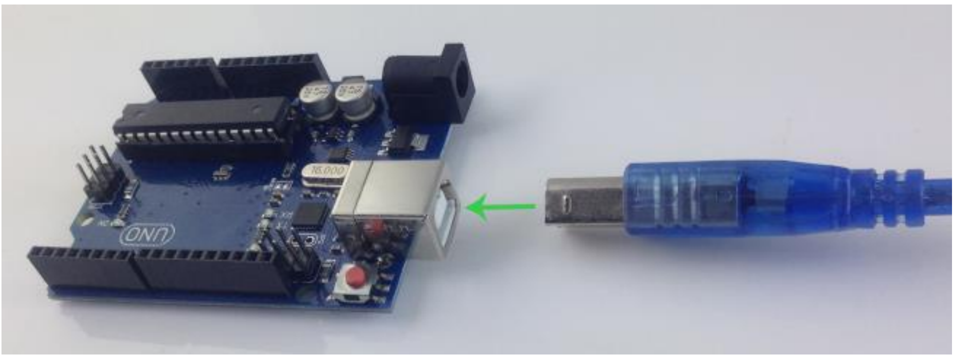
Step 2: Get an Uno R3 and USB cable

In this tutorial, you're using an Uno R3. You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example.



Step 3: Connect the board

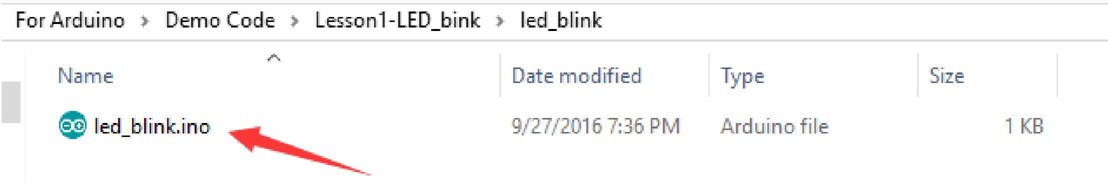
The USB connection with the PC is necessary to program the board and not just to power it up. The Uno and Mega automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable. **The green power LED (labelled PWR) should go on.**

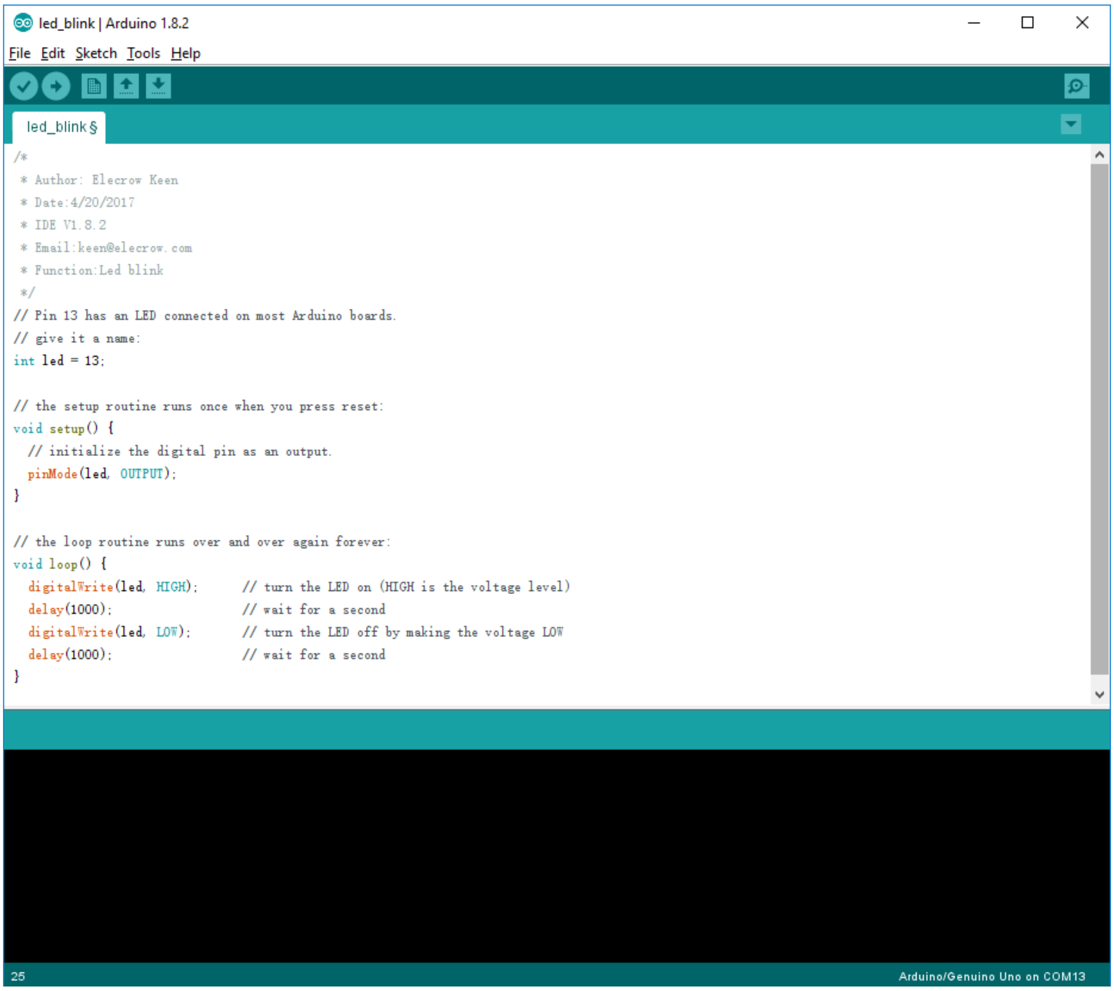




Step 4: Open Lesson 1: LED blink

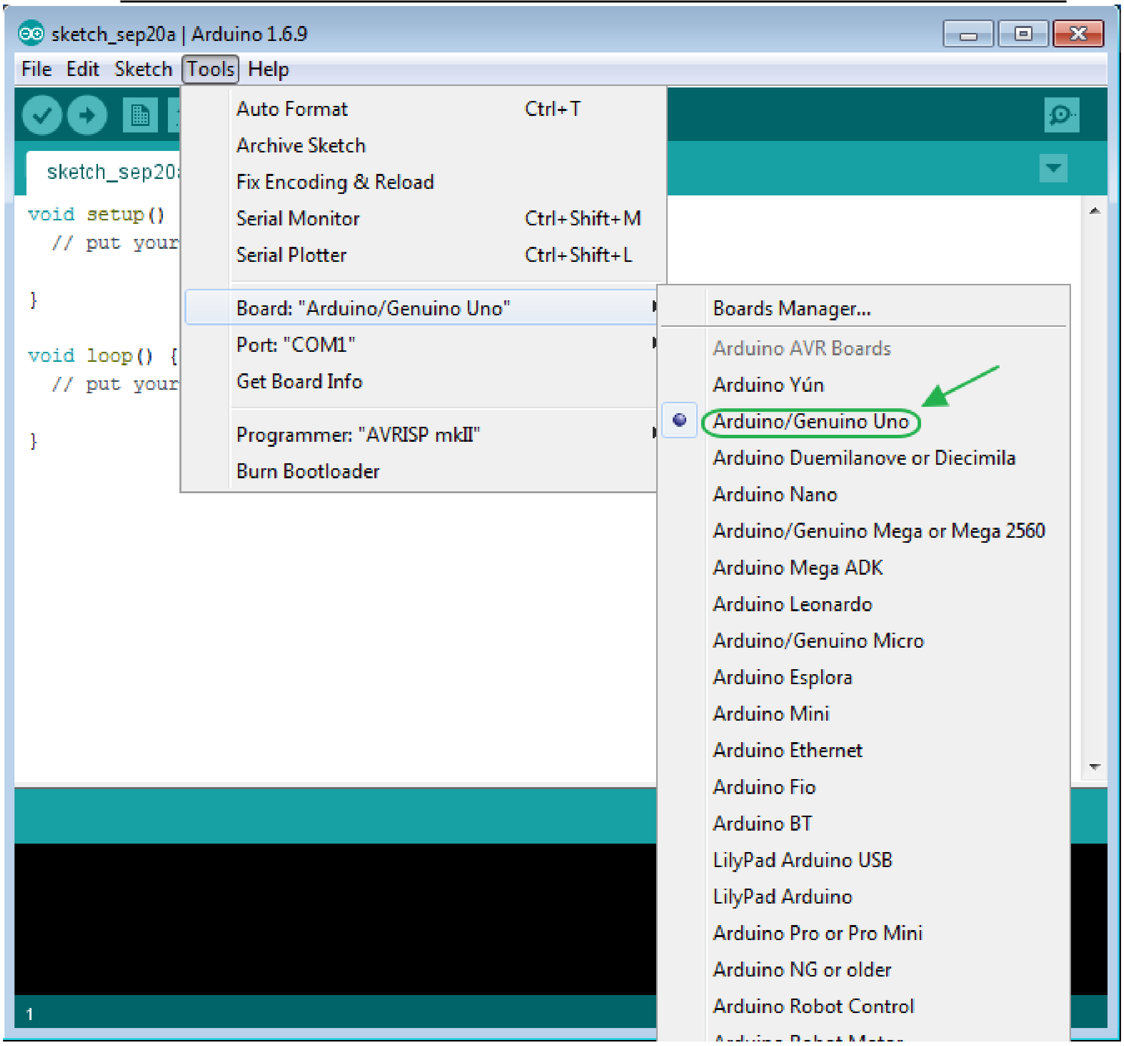
Open the LED blink example sketch: CD > For Arduino>Demo Code>Lesson1-LED_bink>led_blink.





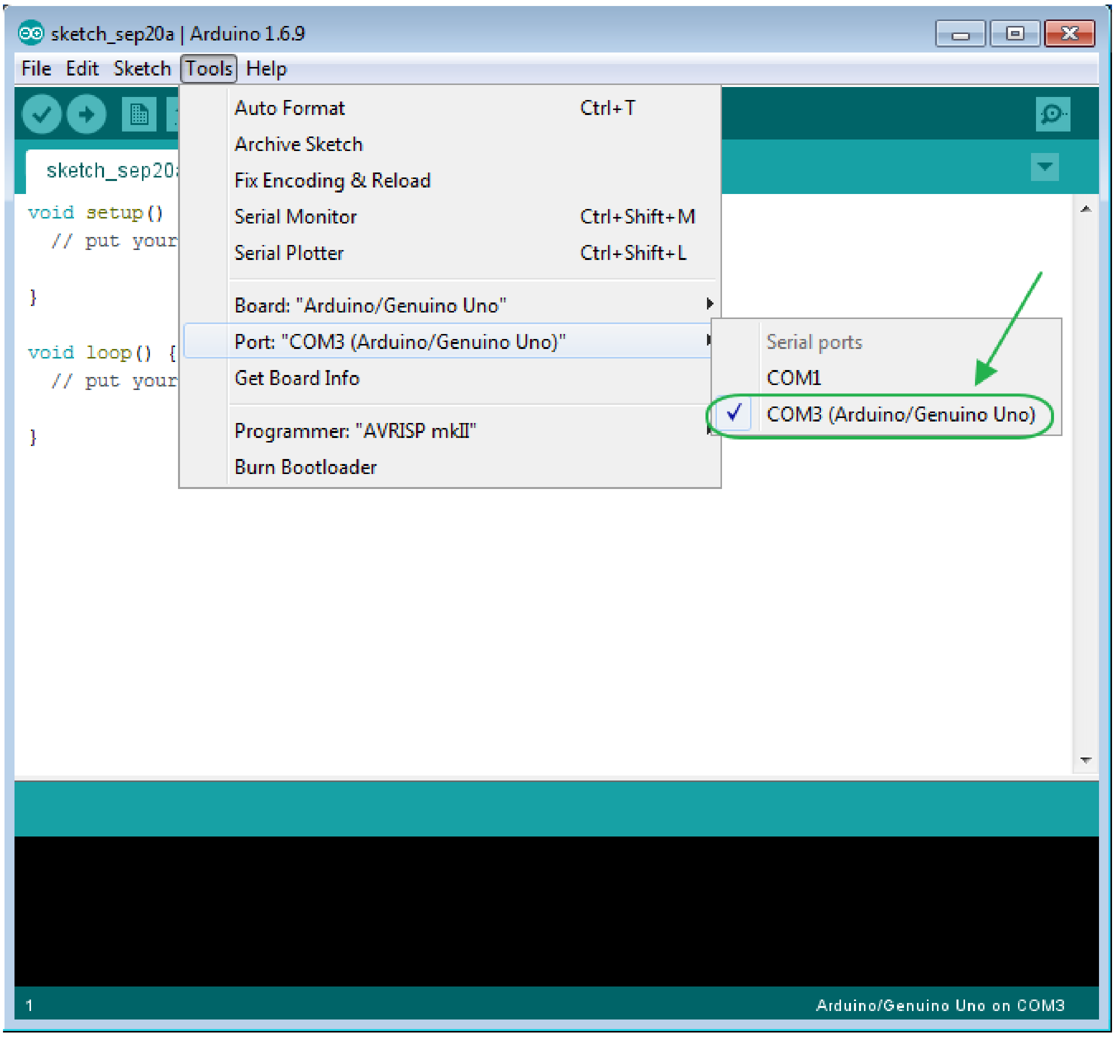
Step 5: Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino board.



Selecting an Arduino/Genuino Uno.

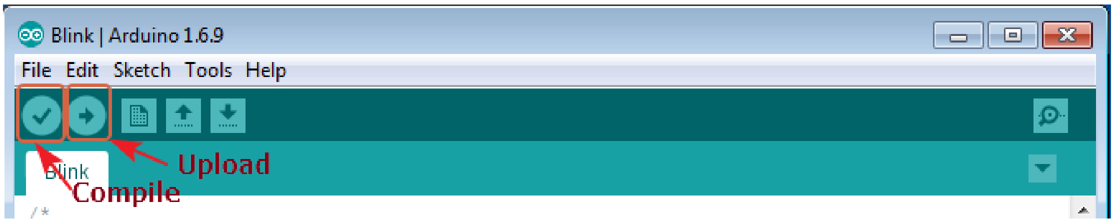
Step 6: Select your serial port



Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 andCOM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Step 7: Upload the program

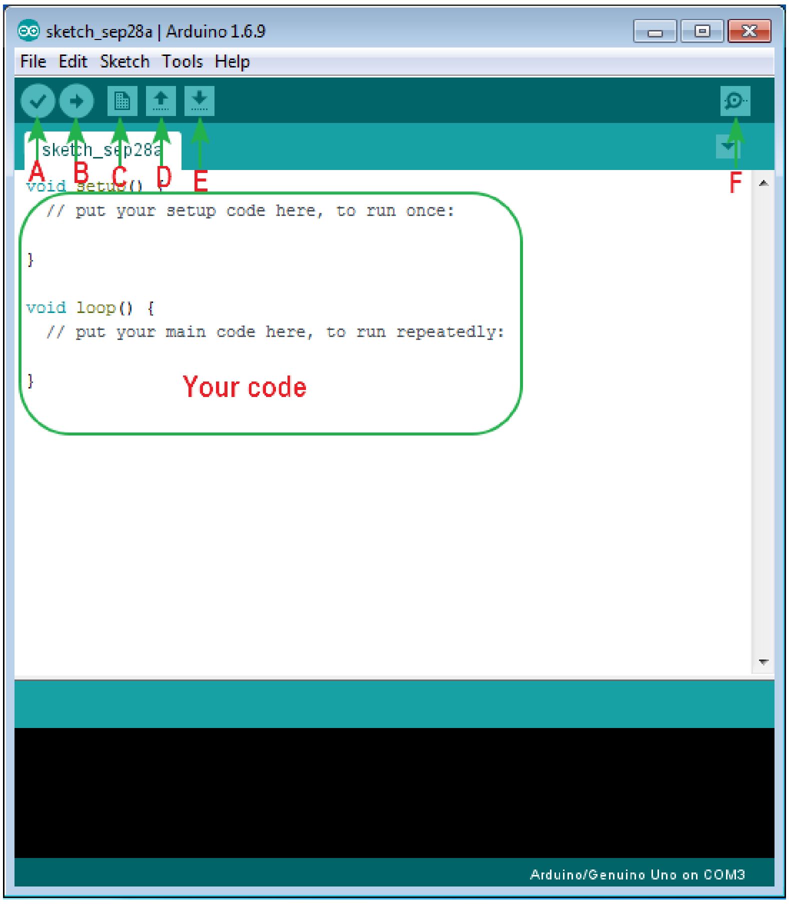
Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



Step 8: Result

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running.

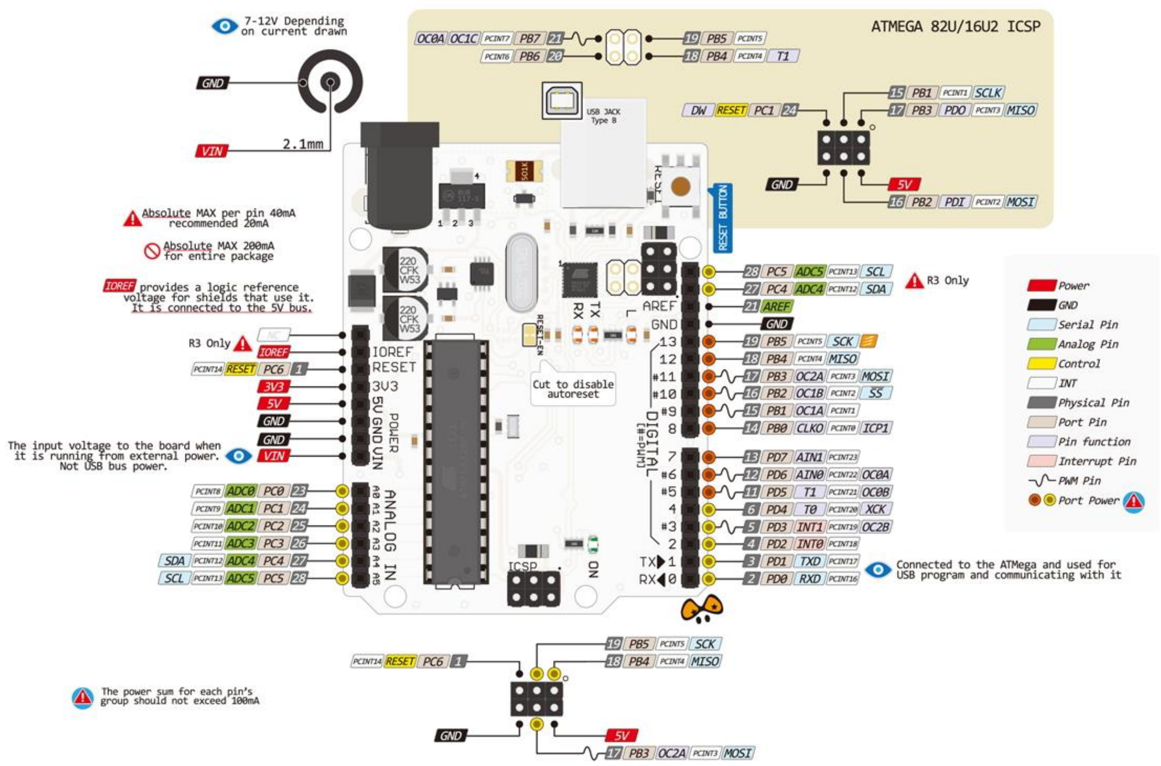
Arduino interface introduction



A ->Compile
B ->Upload
C ->New
D ->Open
E ->Save
F ->Serial monitor

Arduino UNO R3 hardware introduction

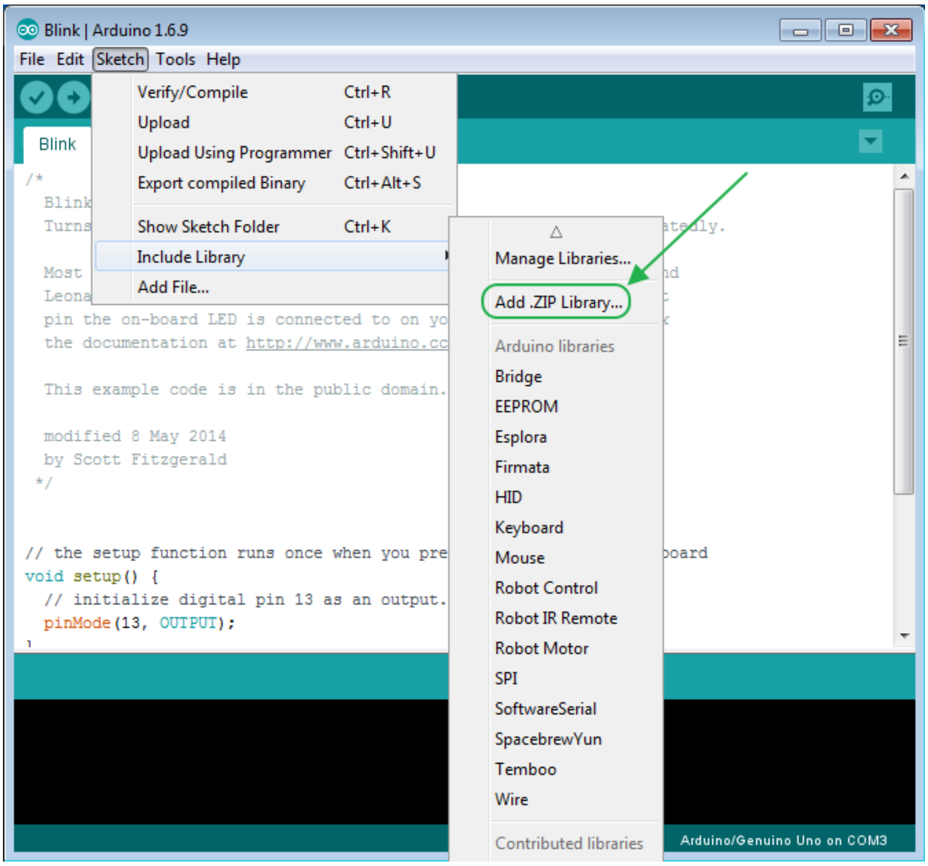




How to add library files

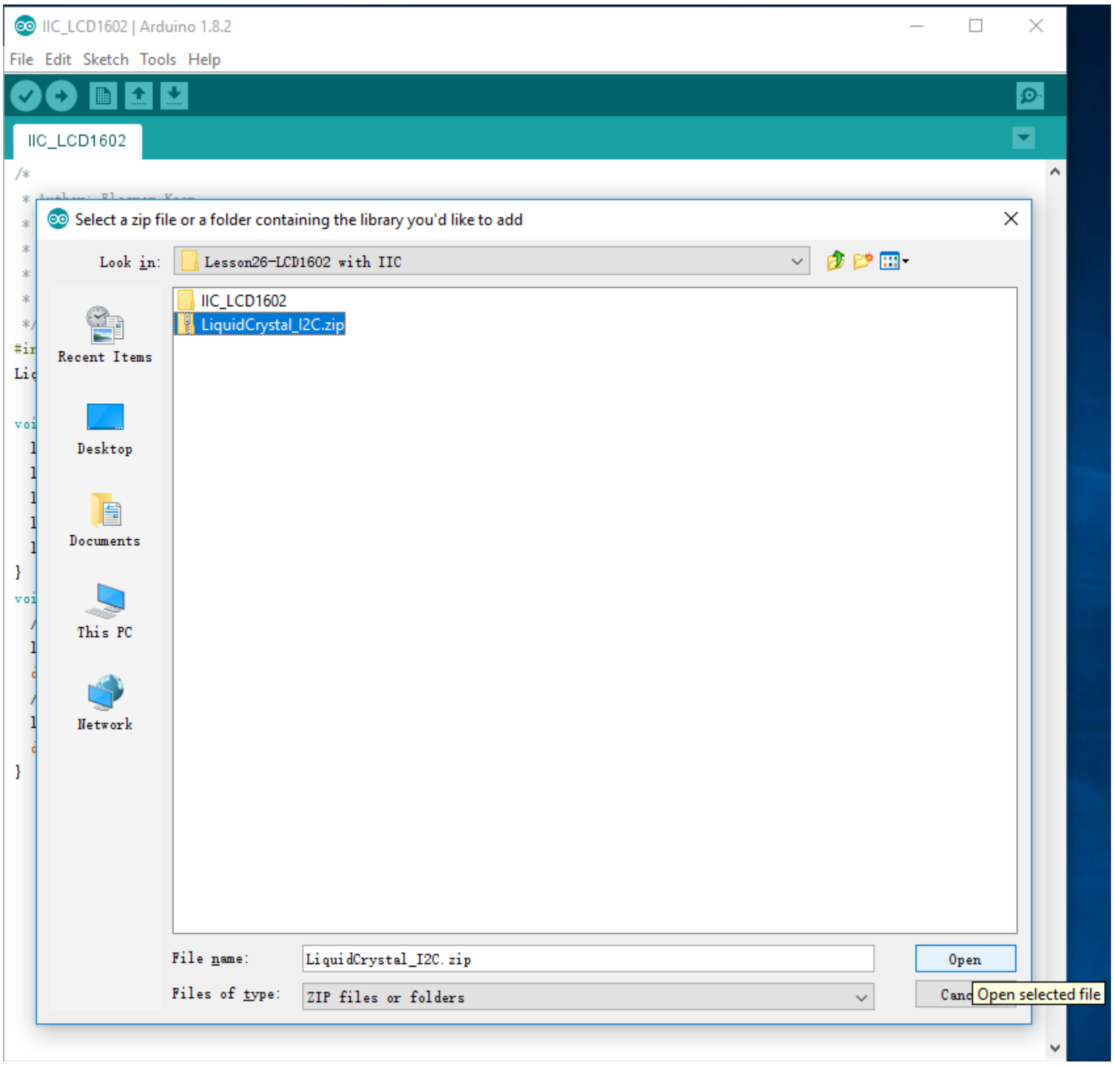
Step 1:

Add library file: Sketch>Include Library>Add.ZIP Library



Step 2:

Select your library file compression package on the demo code file, as follows:



Step 3: Finish.

Learning materials

Ebook

Introduction

The E-book about Arduino what we provided for you is carefully selected and comprehensive, it specially aims at solving the problems when you make projects such as syntax analysis, program optimization and so on. If you have any questions about the projects what we provided, you can also refer the content of e-books.

Path: \For Arduino \Ebook

Language Reference

https://www.elecrow.com/wiki/

http://wiring.org.co/reference/

https://www.arduino.cc/en/Reference/HomePage/

Lessons

Introduction

We will provide you not only the all involved courses about this kit but also to analyze each course. We sincerely hope that you can learn from the first course to the last course because it will lead you start with Arduino step by step, and it also let you jump from a newbie to a higher level for developing your own independent projects.

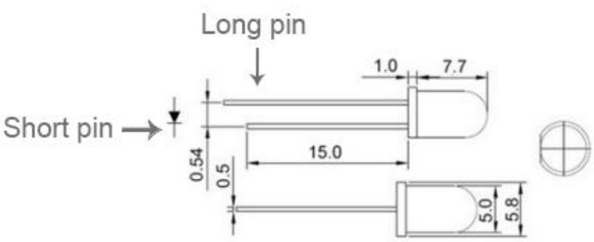
Lesson 1: LED blink

Overview



The LED is designed for the beginners of Arduino. It is the best way to step into the from RPI what it's I/O pins. The LED is the best choice to help you learn I/O pins.

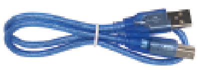
Specification



Pin definition

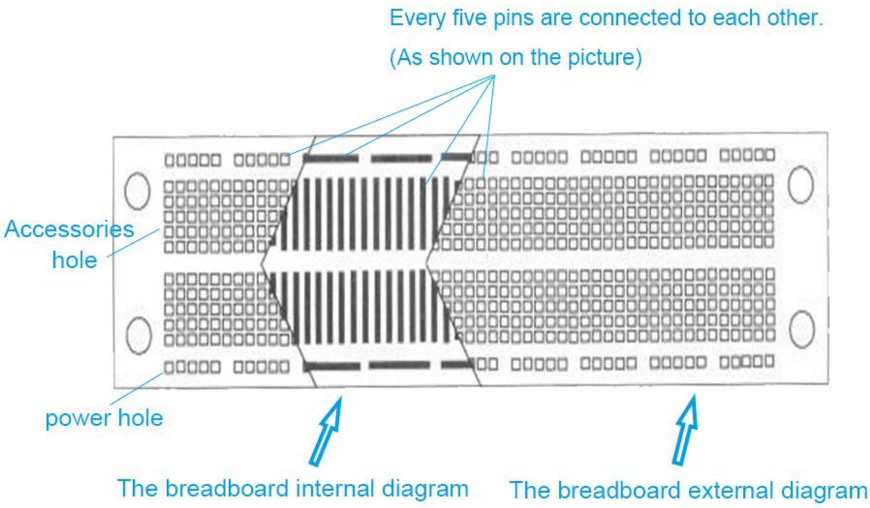
LED		UNO R3
Long pin	->	+5V
Short pin	->	GND

Hardware required

Material diagram	Material name	Number
	220/330Ω resistor	1
	LED	1
	USB Cable	1

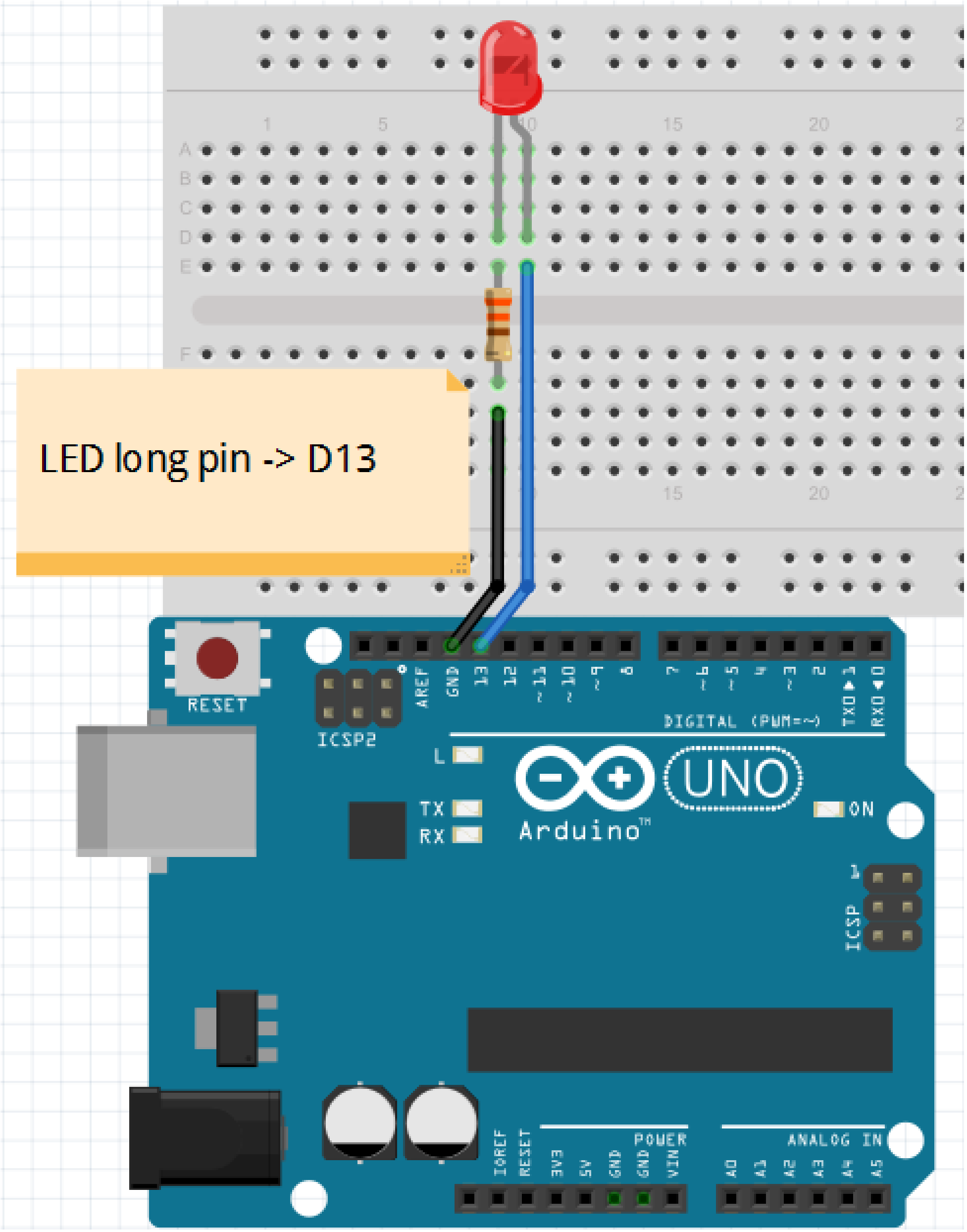
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Bread board schematic



All the tie points (indicated in the picture) of the different colors are connected together.

Connection diagram



Note：The longest LED of the pin is connected to the digital signal port 13(D13).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.

If you fail to open, use the Adobe reader to open this document.
int
setup()
pinMode()
OUTPUT
loop()
HIGH
LOW
digitalWrite()
digitalRead()
delay()
; (semicolon)
{} (curly braces)
= (assign)
// (comment)

Application effect

Turns on an LED on for one second, then off for one second, repeatedly.

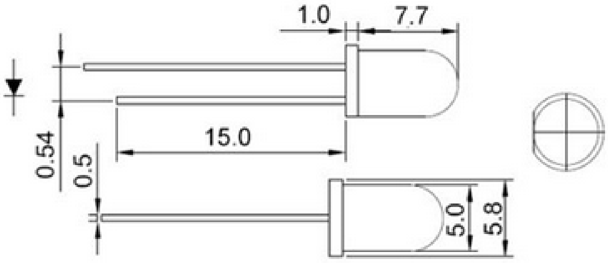
Lesson 2: LED trailing effects

Overview



This lesson will teach you how to show 6 LED trailing effects.

Specification



Pin definition

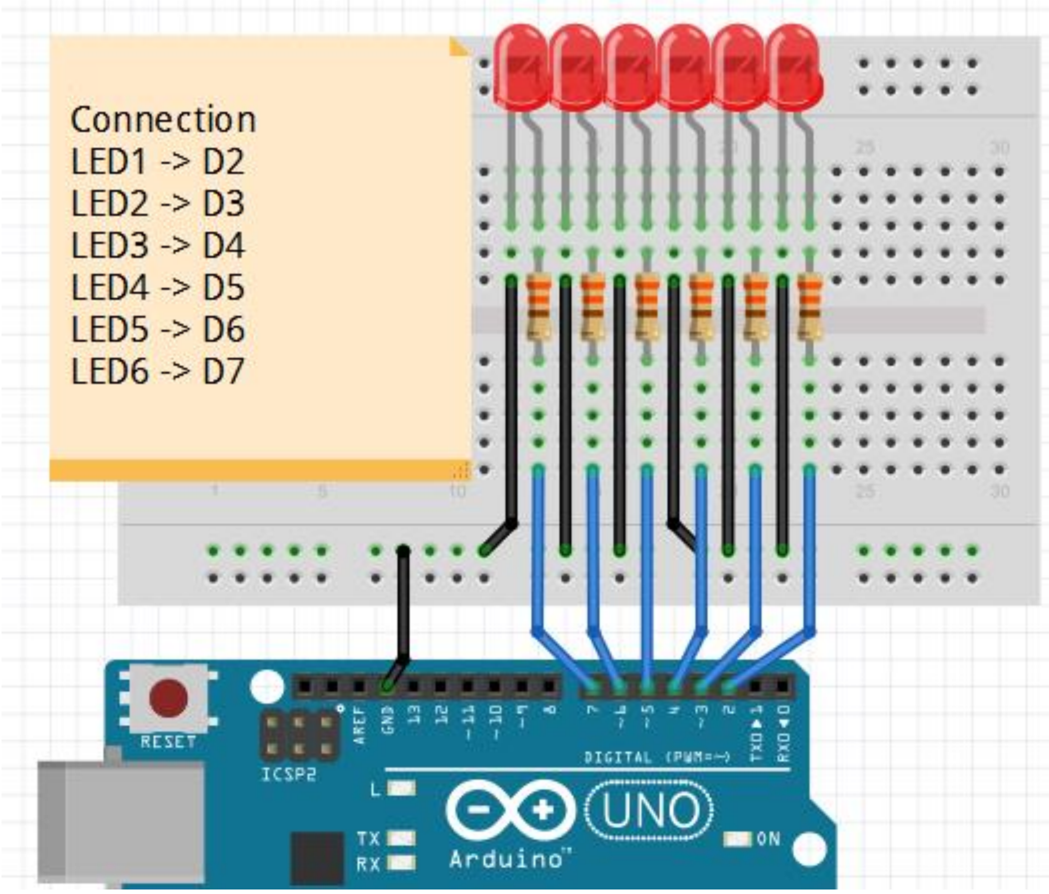
LED UNO R3

LED		UNO R3
Long pin	->	+5V
Short pin	->	GND

Hardware required

Material diagram	Material name	Number
	220/330Ω resistor	6
	LED	6
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note：The longest LED of the pin is connected to the digital signal port *(D*).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.

int
pinMode()
OUTPUT
for()
HIGH
LOW
digitalWrite()

[delay()](#)

Application effect

You'll see all the LEDs will turn on/off regularly.

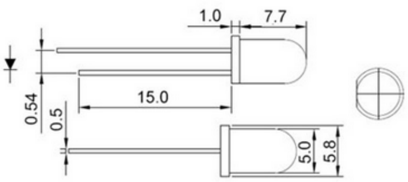
Lesson 3: Traffic light

Overview



The experiment shows the effect of the simulation of traffic lights.

Specification



Pin definition

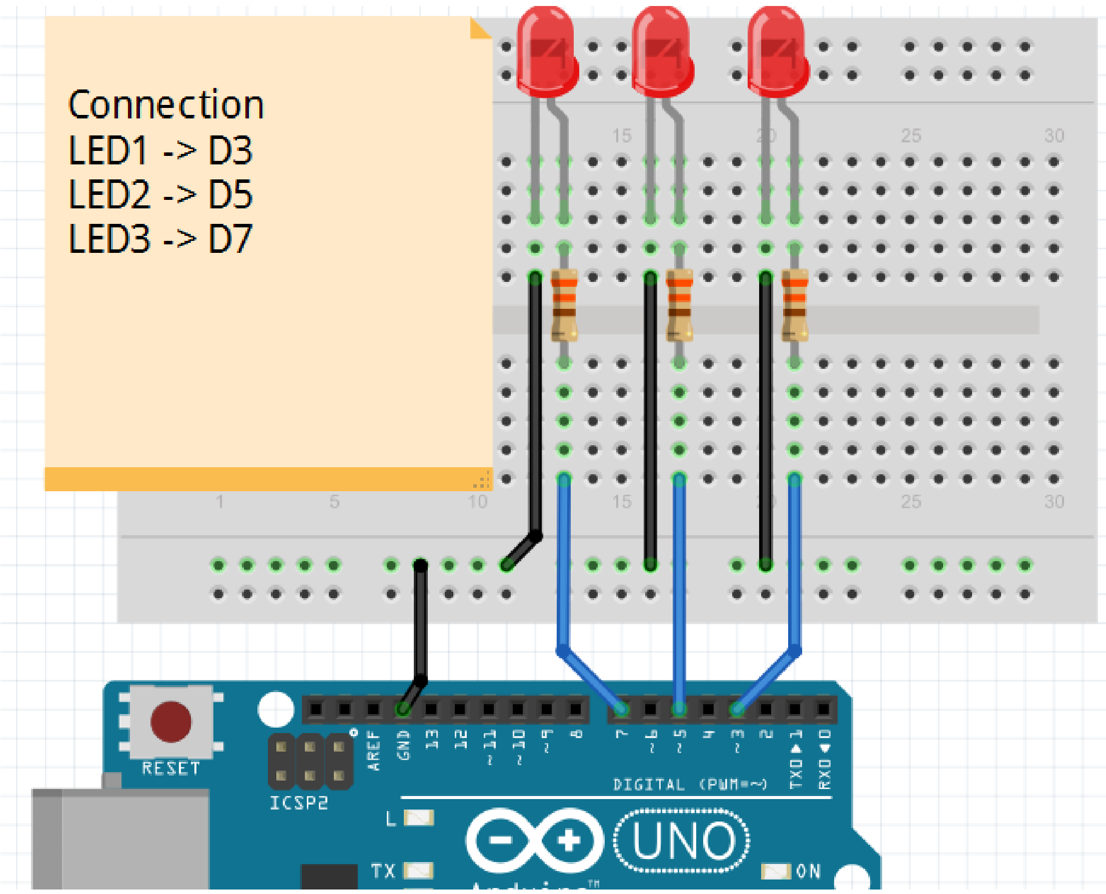
LED UNO R3
Long pin -> +5V
Short pin -> GND

Hardware required

Material diagram	Material name	Number
	220/330Ω resistor	3

	Yellow LED	1
	Green LED	1
	Red LED	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note：The longest LED of the pin is connected to the digital signal port *(D*).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.

pinMode()
OUTPUT
INPUT
for()
HIGH
LOW
digitalWrite()
delay()
< (less than)
++ (increment)

Application effect

The green light flashes for 5 seconds, then the yellow light flashes 3 times, and then the red light 5 seconds, the formation of a cycle. And then repeat the cycle. This experiment shows the effect of the simulation of traffic lights.

Lesson 4: Analog input

Overview



In this lesson, we use a variable resistor (a potentiometer), we read its value using one analog input of an Arduino board and we change the blink rate of the built-in LED accordingly. The resistor's analog value is read as a voltage because this is how the analog inputs work.

Specification

Product Name: Potentiometer;

Resistance Value: 10K ohm;

Adjustment Type: Top Adjustment

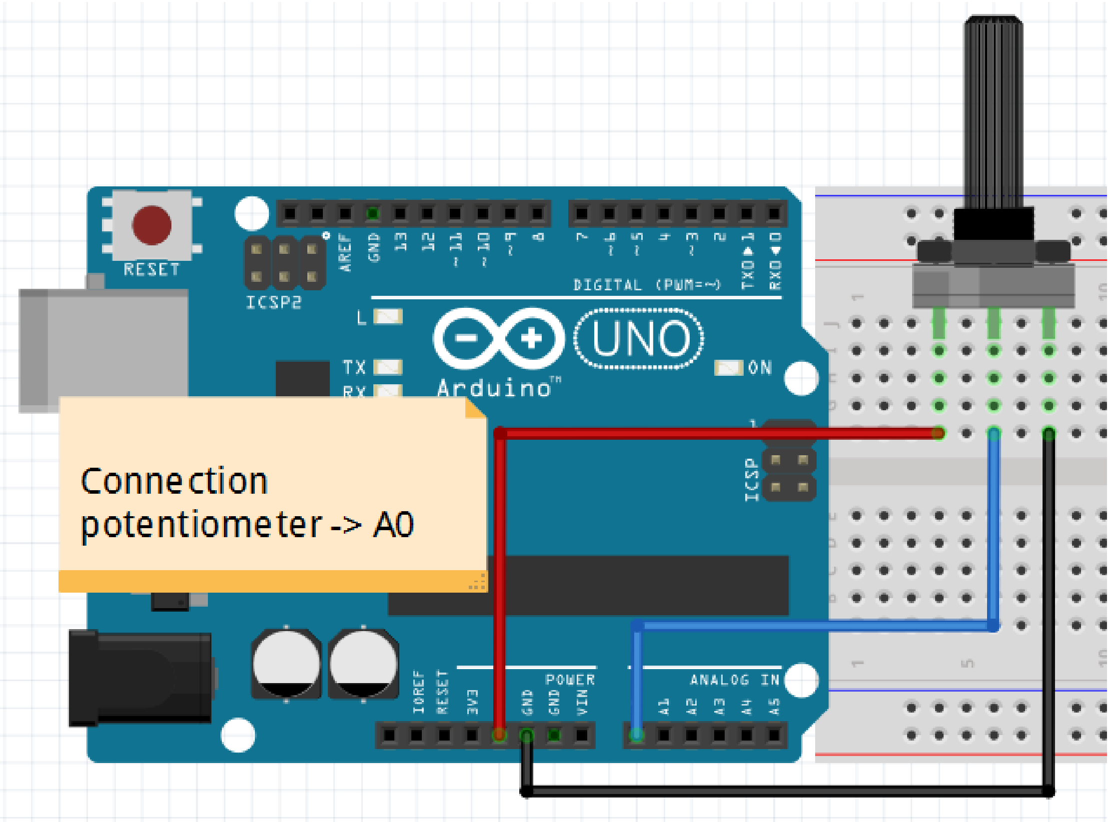
Pin definition

Null

Hardware required

Material diagram	Material name	Number
	10KΩ potentiometer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note：The middle pin of the potentiometer is connected to the analog port 0(A0).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
digitalWrite()
analogRead()

Application effect

By turning the shaft of the potentiometer, you change the amount of resistance on either side of the center pin (or wiper) of the potentiometer. This changes the relative resistances between the center pin and the two outside pins, giving you a different voltage at the analog input. When the shaft is turned all the way in one direction, there is no resistance between the center pin and the pin connected to ground. The voltage at the center pin then is 0 volts, and analogRead() returns 0. When the shaft is turned all the way in the other direction, there is no resistance between the center pin and the pin connected to +5 volts. The voltage at the center pin then is 5 volts, and analogRead() returns 1023. In between, analogRead() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

That value, stored in sensorValue, is used to set a delay() for your blink cycle. The higher the value, the longer the cycle, the smaller the value, the shorter the cycle. The value is read at the beginning of the cycle, therefore the on/off time is always equal.

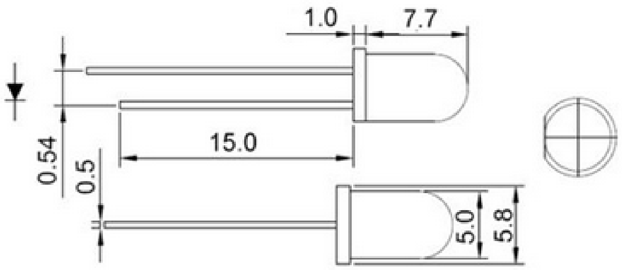
Lesson 5: Fading

Overview



This example demonstrates the use of analog output (Pulse Width Modulation (PWM)) to fade an LED. PWM is a technique for getting an analog-like behavior from a digital output by switching it off and on very fast and with different ratio between on and off time.

Specification



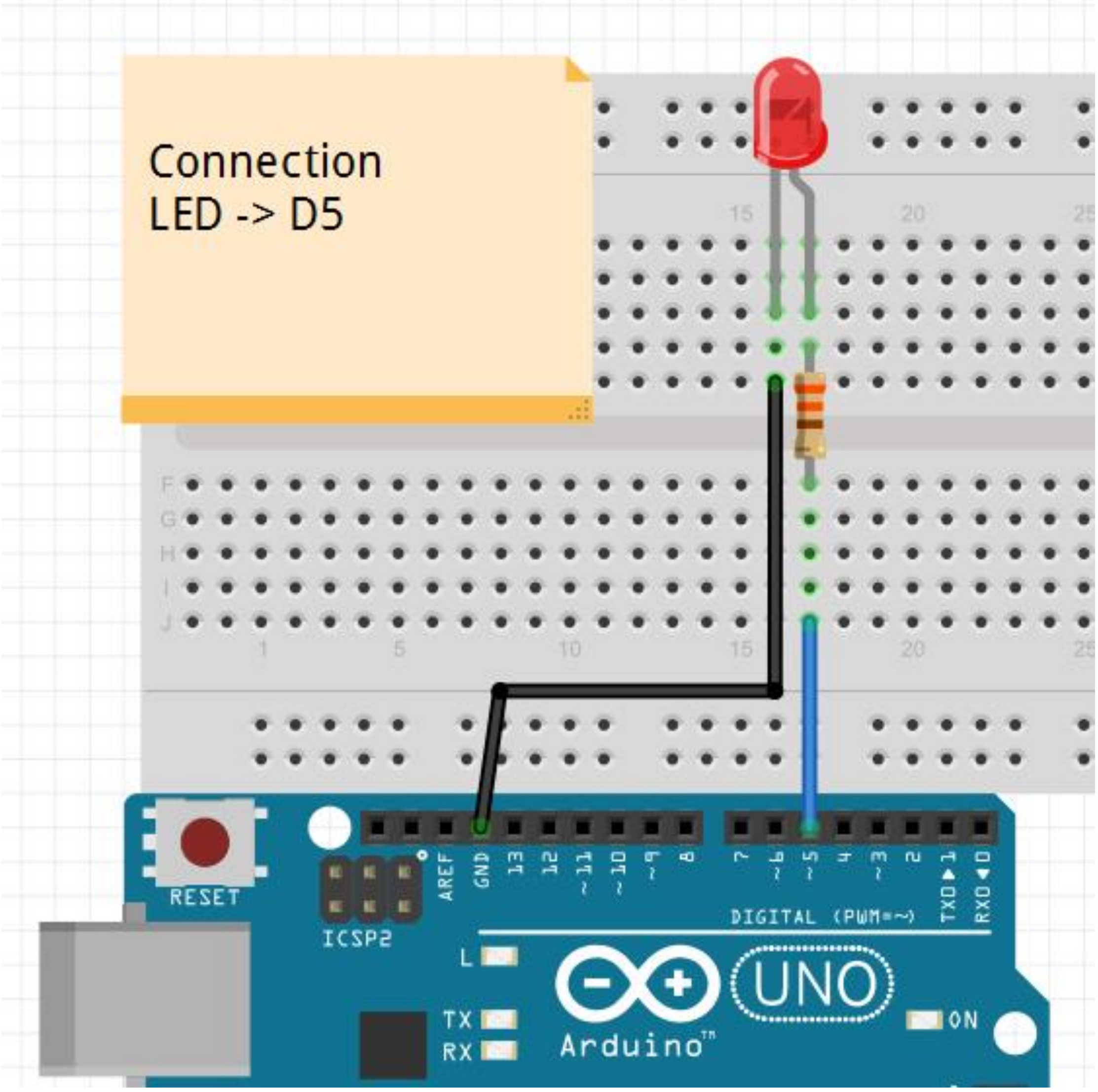
Pin definition

LED		UNO R3
Long pin	->	+5V
Short pin	->	GND

Hardware required

Material diagram	Material name	Number
	220/330Ω resistor	1
	LED	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note：An LED connected to digital output pin 5 (D5) through a 220 ohm resistor.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.

+= (add assign)

-= (subtract assign)

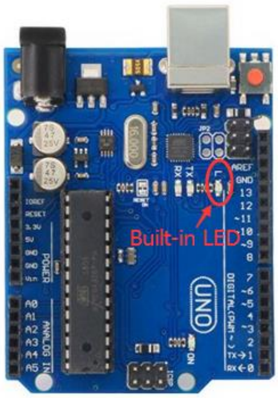
<= (less than or equal to)
>= (greater than or equal to)

Application effect

You'll see that LED has the effect of breathing light.

Lesson 6: Button

Overview

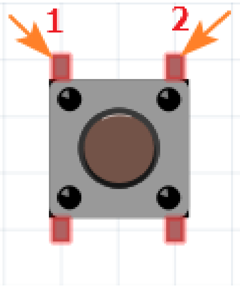
Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Specification

Size: 6 x 6 x 5mm
Temperature: -30 ~ +70 Centigrade

Pin definition

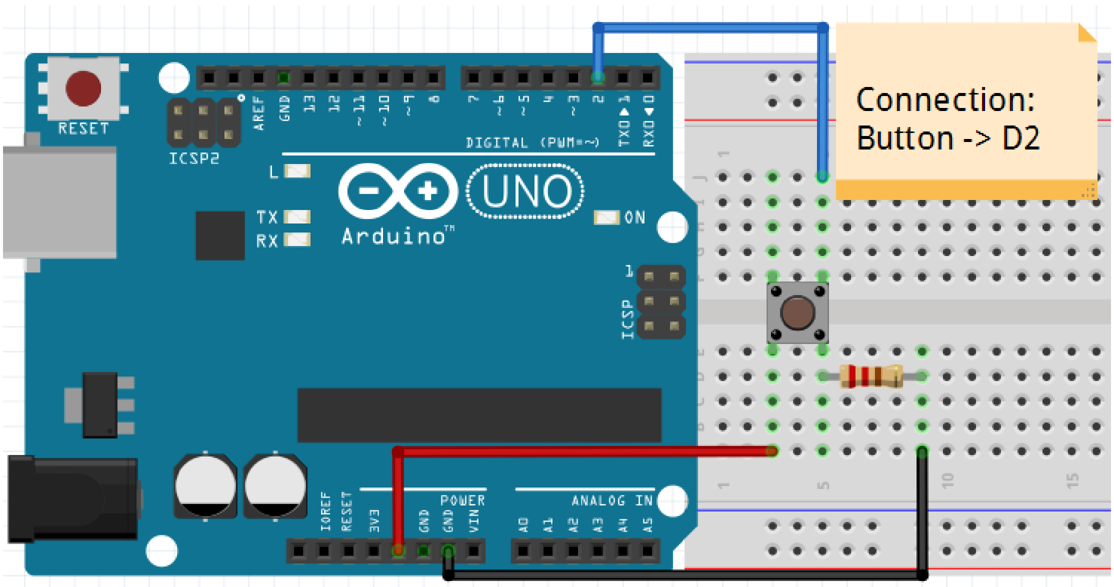
It is the definition of Button pin：



Hardware required

Material diagram	Material name	Number
	Button	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Connect three wires to the board. The first two, red and black, connect to the two long vertical rows on the side of the breadboard to provide access to the 5 volt supply and ground. The third wire goes from digital pin 2 to one leg of the pushbutton. That same leg of the button connects through a pull-down resistor (here 10K ohm) to ground. The other

leg of the button connects to the 5 volt supply.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to 5 volts, so that we read a HIGH.

You can also wire this circuit the opposite way, with a pullup resistor keeping the input HIGH, and going LOW when the button is pressed. If so, the behavior of the sketch will be reversed, with the LED normally on and turning off when you press the button.

If you disconnect the digital I/O pin from everything, the LED may blink erratically. This is because the input is "floating" - that is, it will randomly return either HIGH or LOW. That's why you need a pull-up or pull-down resistor in the circuit.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
const
INPUT

Application effect

When you press the button, the built-in LED will light up, release is extinguished.

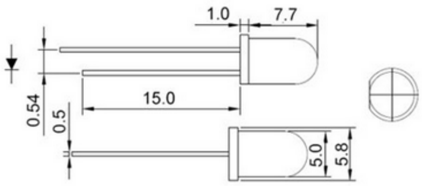
Lesson 7: Responder experiment

Overview



This lesson will teach you how to be a responder.

Specification

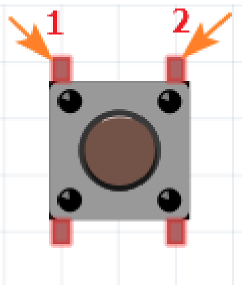


Button : Size: 6 x 6 x 5mm LED:

Temperature: -30 ~ +70 Centigrade

Pin definition

Is the definition of Button pin :

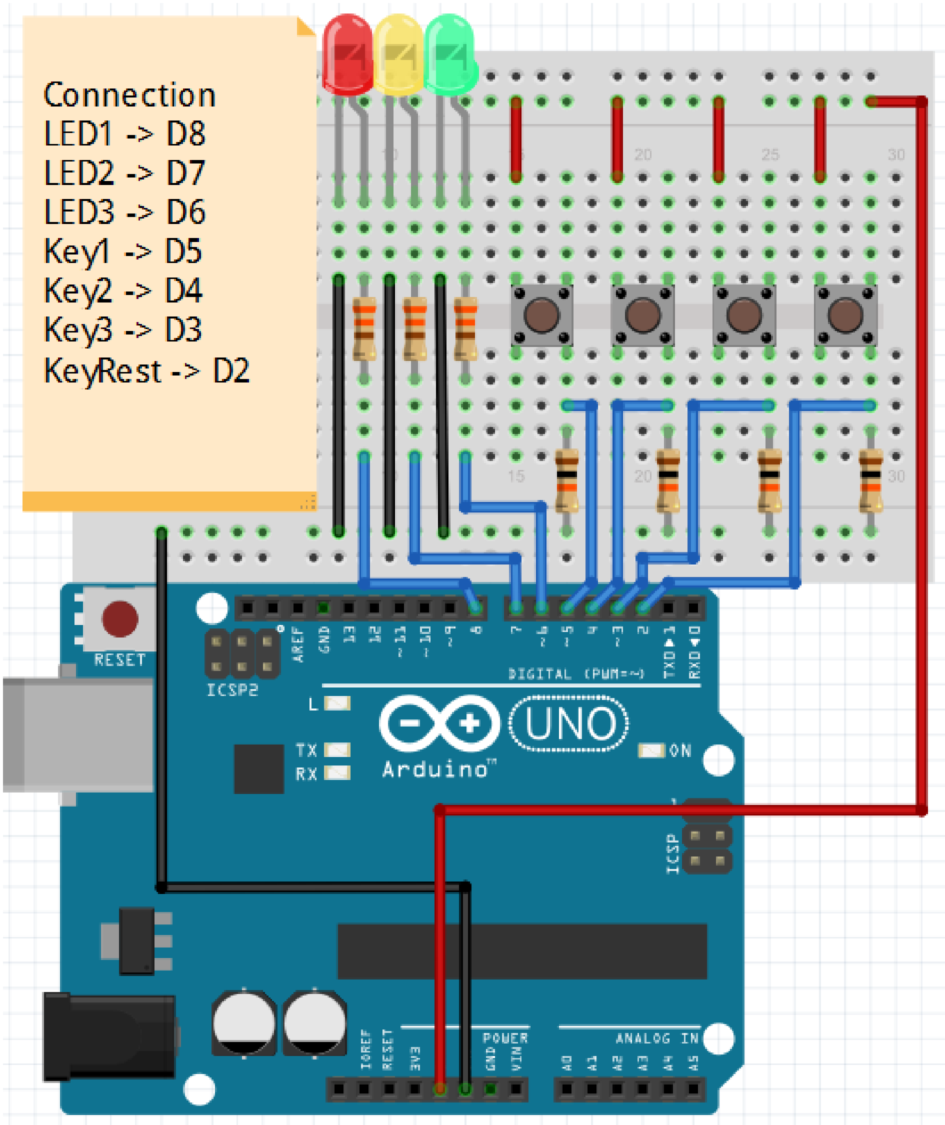


Hardware required

Material diagram	Material name	Number

	Button	4
	LED	3
	220/330Ω resistor	3
	10KΩ resistor	4
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: Button using 10KΩ resistor, LED use 220/330Ω resistor.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
digitalRead()
== (equality)

Application effect

Whichever button is pressed first, then the corresponding LED will be on!
If you want to reset, hit the Reset button.

Lesson 8: Active buzzer

Overview



This is an active buzzer experiment. Active means that the direct power supply can make a sound.

Specification

Voltage: DC 5V
Min Sound Output at 10cm: 85dB;
Total Size (Pin Not Included): 12 x 9mm/0.47" x 0.35"(D*H)

Pin definition

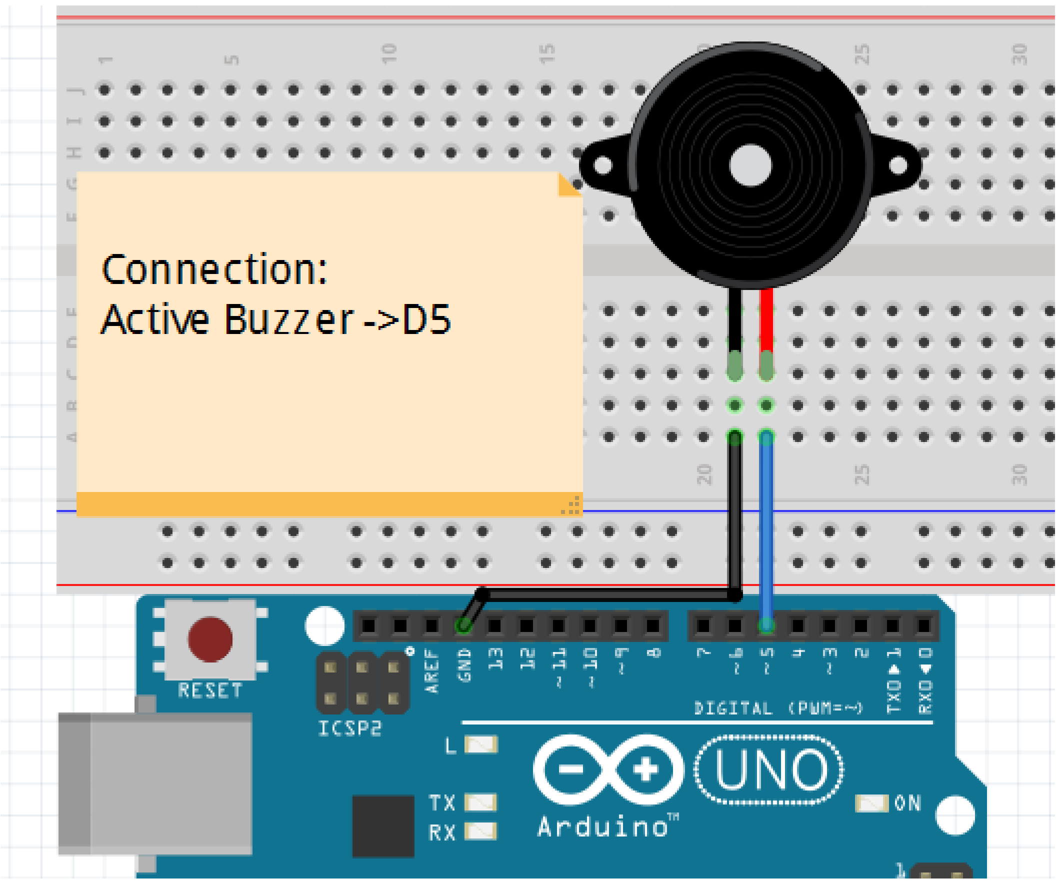
Active Buzzer UNO R3

Long pin -> D5
Short pin -> GND

Hardware required

Material diagram	Material name	Number
	Active buzzer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note：The longest active buzzer of the pin is connected to the digital signal port 5 (D5).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
digitalWrite()
pinMode()

Application effect

When the upload process is complete, the buzzer rings.

Lesson 9: Passive buzzer

Overview



Specification

Working Voltage: 3V/5V
Resistance: 16Ohm
Resonance Frequency: 2KHZ

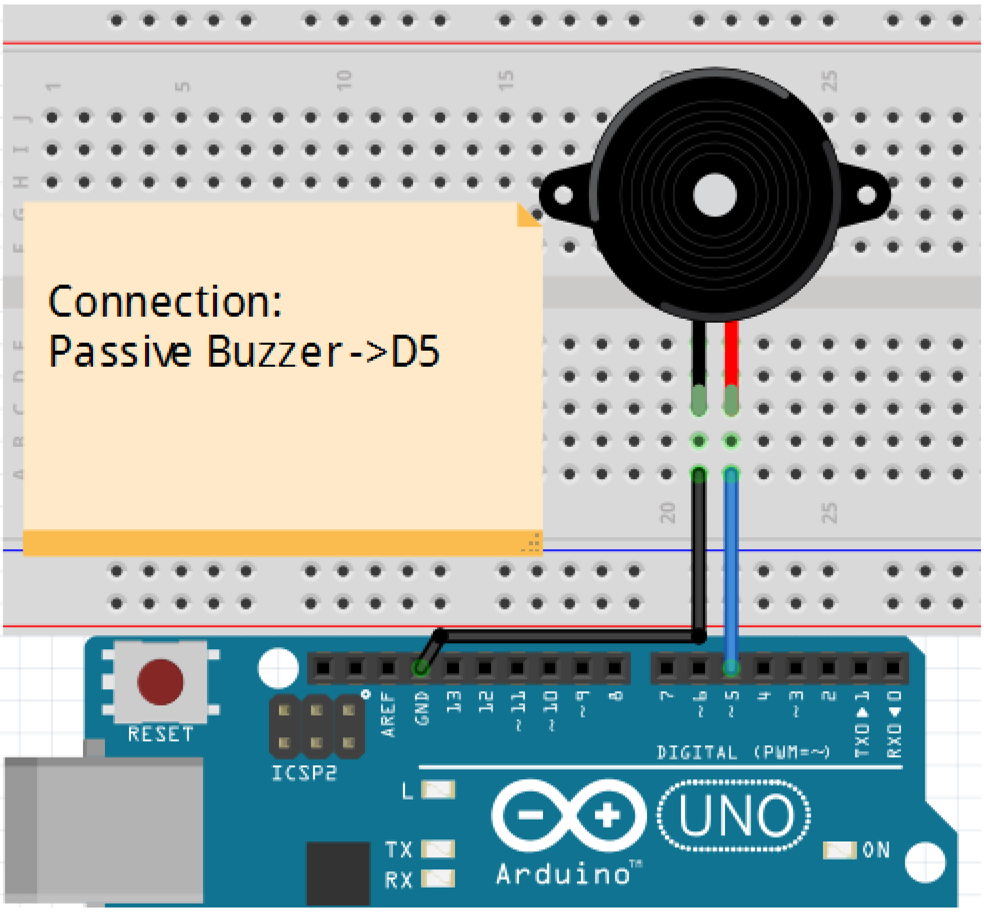
Pin definition

Passive Buzzer		UNO R3
Long pin	->	D5
Short pin	->	GND

Hardware required

Material diagram	Material name	Number
	Passive buzzer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
#define
tone()

Application effect

When the upload process is complete, the buzzer sounds for 2 seconds.

Lesson 10: RGB LED

Overview



In this lesson, you will learn how to use a RGB (Red Green Blue) LED with an Arduino. You will use the analogWrite function of Arduino to control the color of the LED.

Specification

Emitting Light Color: Blue, Red, Green

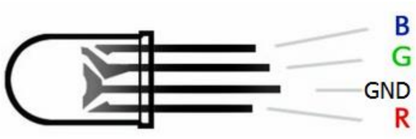
Size(Approx): 5 x 35mm/ 0.2" x 1.37" (D * L)

Forward Voltage: 3.0-3.4V

Luminous Intensity: 12000-14000mcd

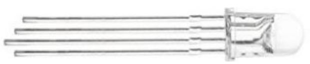
Pin definition

It is the definition of RGB LED pin：

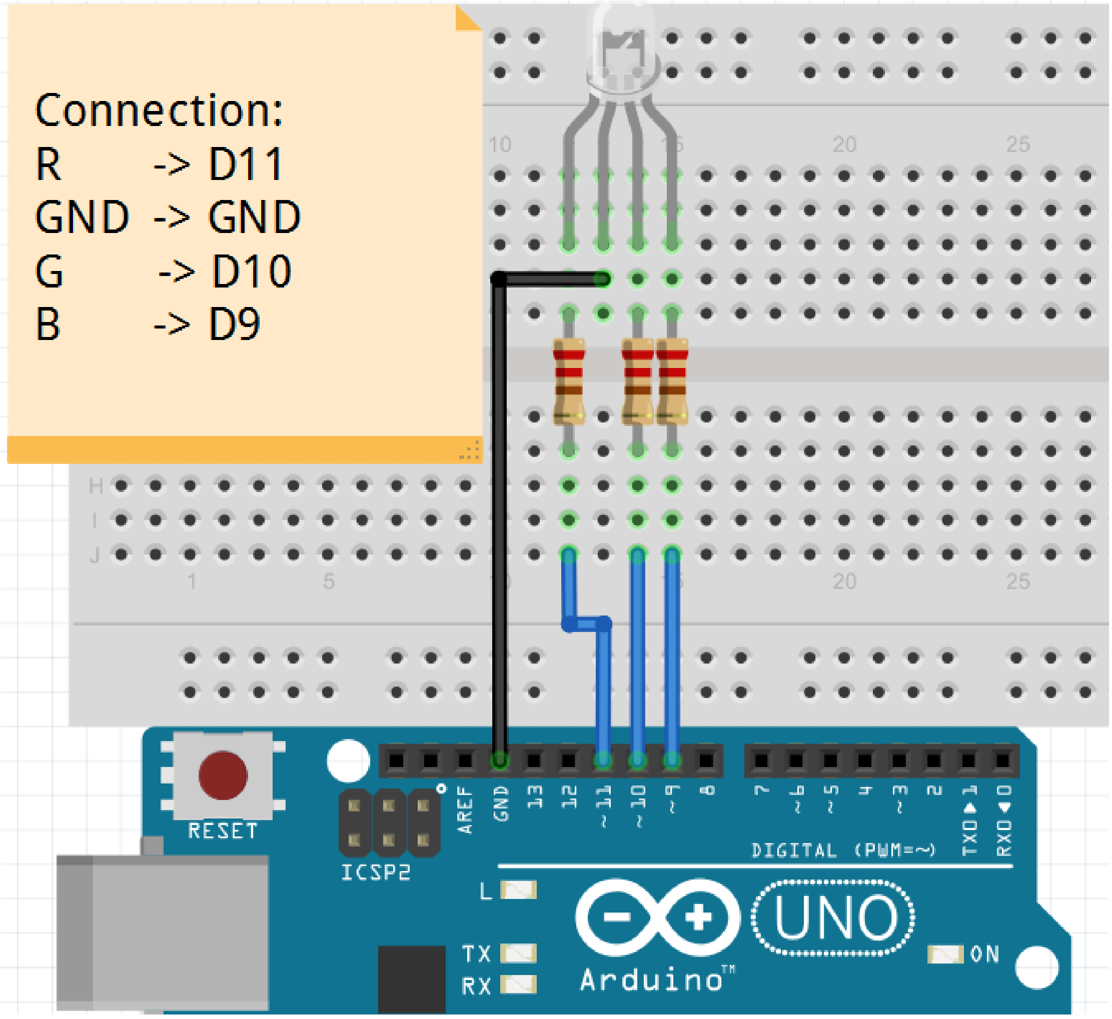


RGB LED		UNO R3
R	->	D11
GND	->	GND
G	->	D10
B	->	D9

Hardware required

Material diagram	Material name	Number
	RGB LED	1
	220Ω/330Ω resistor	3
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: The longest pin of the RGB LED is connected to the GND.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips: click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
analogWrite()
#define

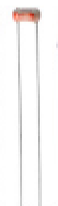
Application effect

When the program is uploaded, you will see the LED loop emit 7 different colors of light.

Lesson 11: Making sounds

Overview





In this lesson, you will learn how to make sounds with your Arduino. First you will make the Arduino play a 'musical' scale and then combine this with a photocell, to make a Theremin-like instrument that changes the pitch played as you wave your hand over the photocell.

Specification

Passive buzzer:
Working Voltage: 3V/5V
Resistance: 16Ohm
Resonance Frequency: 2KHZ

Photoresistor:
Model: GL5528
Maximum Voltage: 150 Volt DC
Spectral Peak: 540nm
Maximum Wattage: 100mW
Operating Temperature: -30 ~ +70℃
Light Resistance (10 Lux): 10-20Kohm

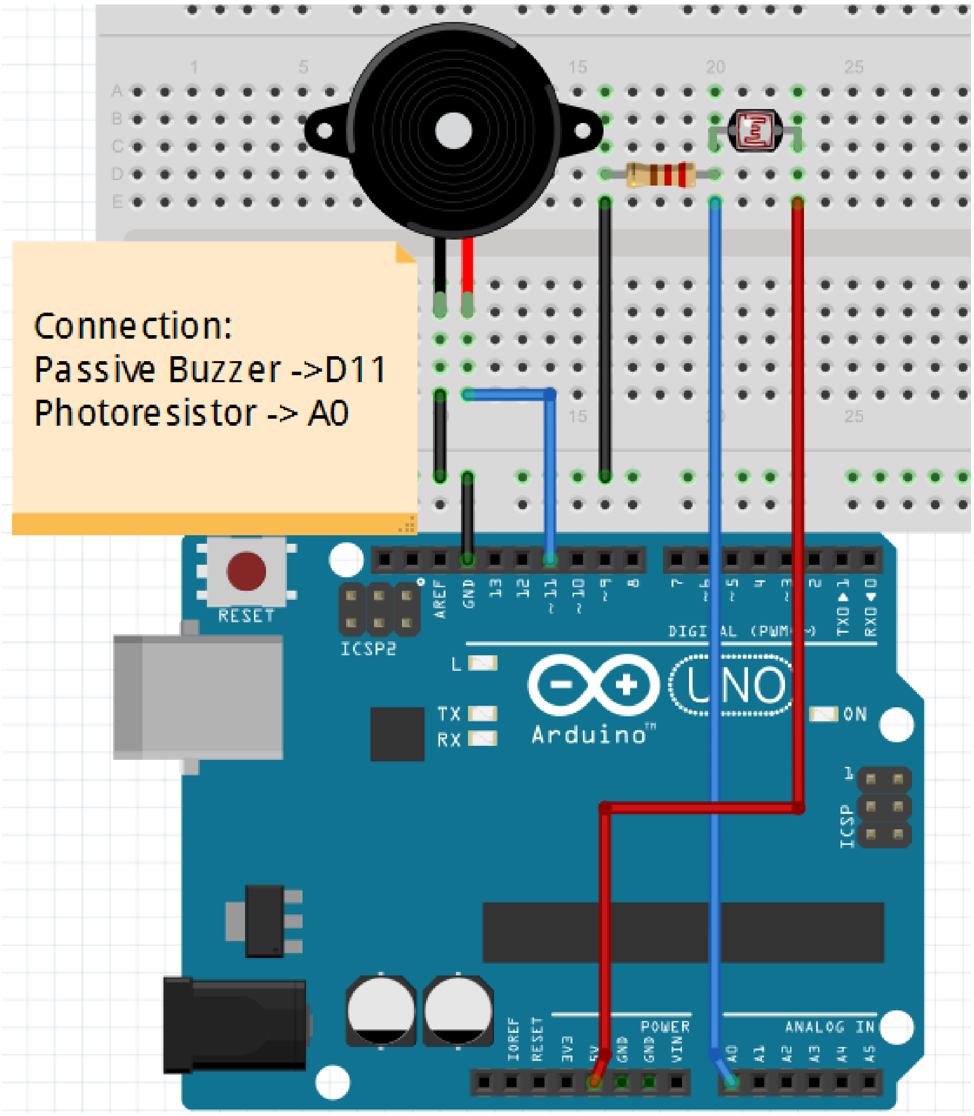
Pin definition

Passive Buzzer		UNO R3
Long pin	->	D11
Short pin	->	GND

Hardware required

Material diagram	Material name	Number
	Photoresistor	1
	Passive buzzer	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: Photoresitor`s pin is not divided into positive and negative polarity

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
Try changing the value 4 in the line below to lower and higher values.
//int pitch = 200 + reading / 4;
We simply take an analog reading from A0, to measure the light intensity. This value will be in the range of something like 0 to 700.
We add 200 to this raw value, to make 200 Hz the lowest frequency and simply add the reading divided by 4 to this value, to give us a range of around 200Hz to 370Hz.

Language reference

tone()
+ (addition)
/ (divide)

Application effect

When you use the hand slowly close to the photosensitive resistance, the buzzer sounds will be changed.

Lesson 12: Analog temperature

Overview



This lesson we will teach you how to read the value of the thermistor.

Specification

Model: MF52-103
Insulation Material: Ceramic

Color: Black
Rated Power: 0.05W
Resistance Value: 10k
Resistance Tolerance: H (Â±3%)
B Value: 3950K
Pin Pitch: 1.5mm / 0.059"

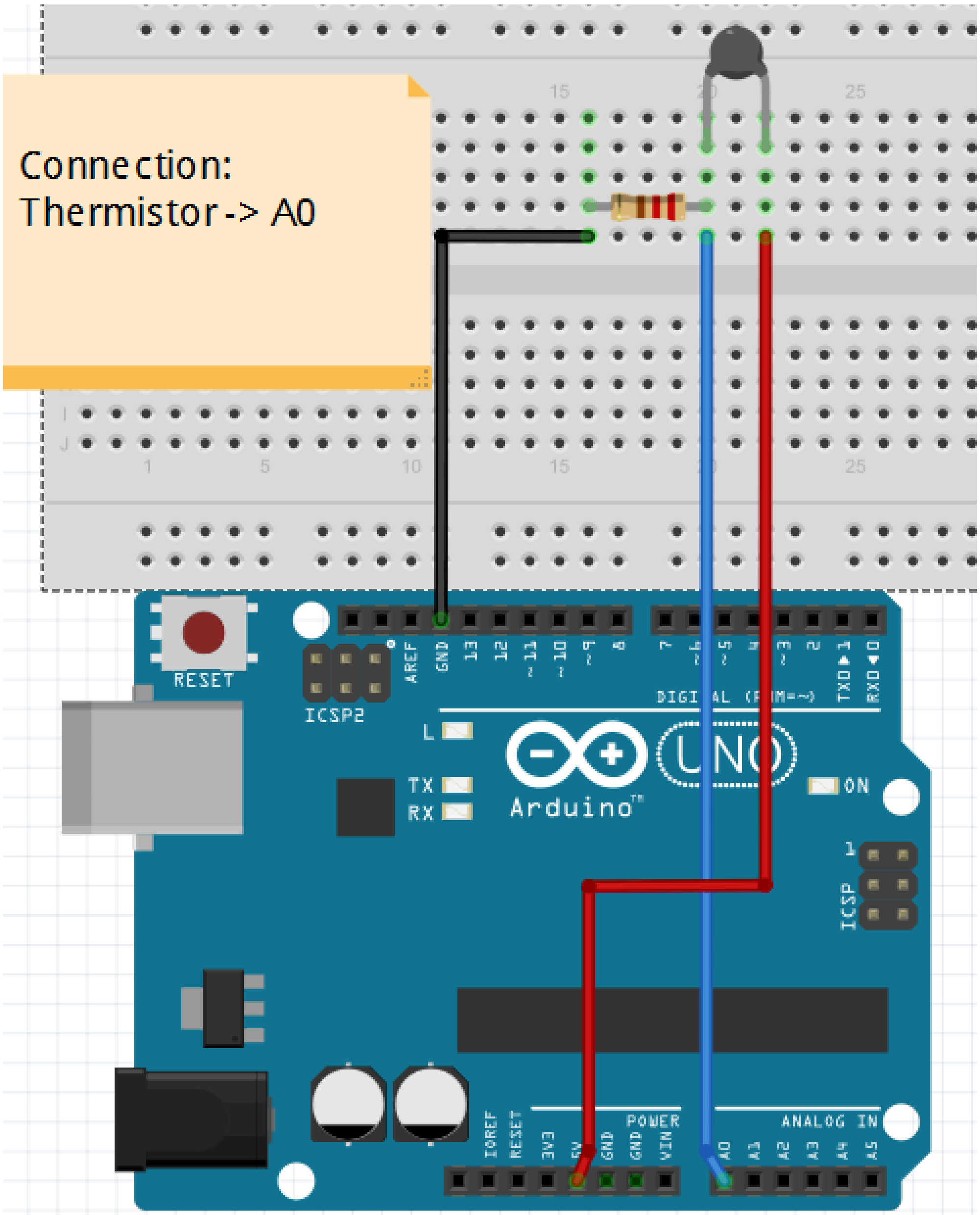
Pin definition

Nonpolar

Hardware required

Material diagram	Material name	Number
	Thermistor	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

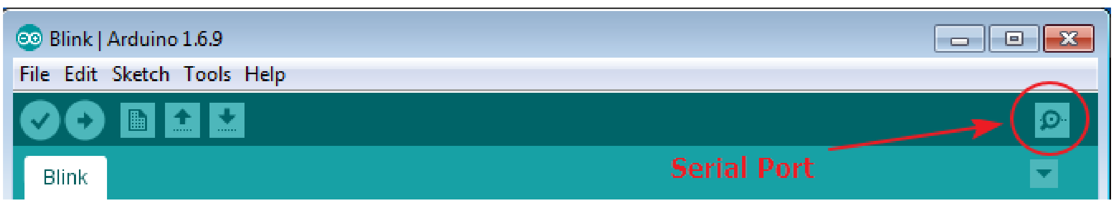
Connection diagram



Note: Thermistor `s pin does not distinguish between positive and negative poles.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8). And open the serial port.



Language reference

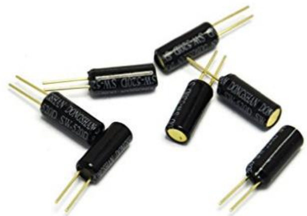
[serial](#)
[DEC](#)

Application effect

After uploading the program, open the serial port monitor, you will see a series of temperature values

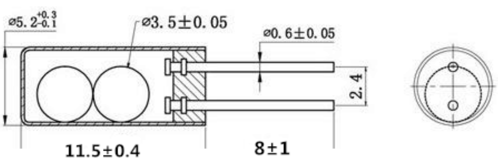
Lesson 13: Tilt switch

Overview



This is a very simple switch experiment.

Specification



Pin definition

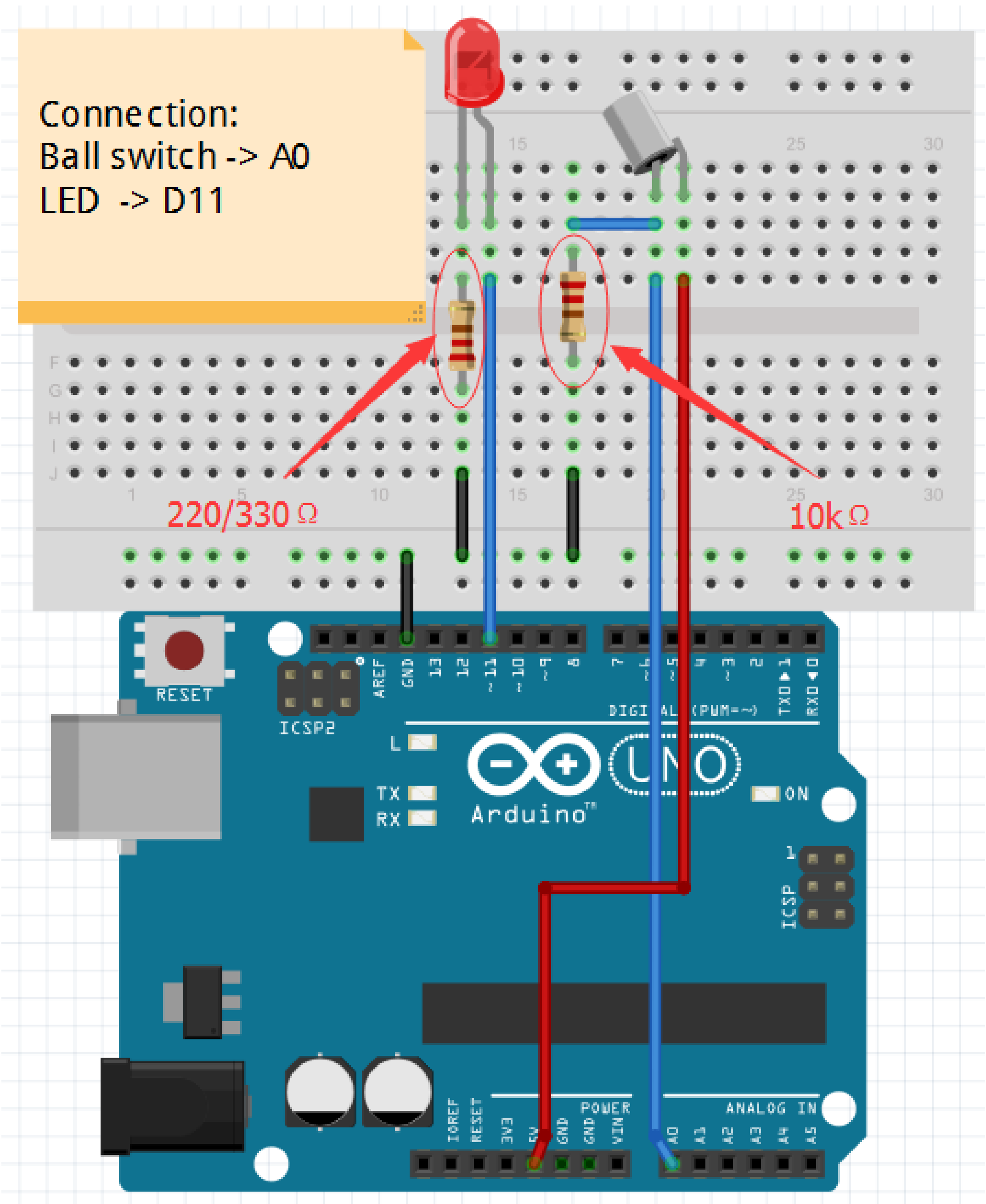
Nopolarity.

Hardware required

Material diagram	Material name	Number
	Ball switch	1
	LED	1

	220/330Ω resistor	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: The longest LED of the pin is connected to the digital signal port 11 (D11).
Ball switch`s pin is not divided into positive and negative polarity.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

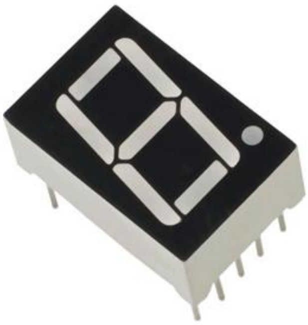
Tips: click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
If()
else

Application effect

LED light up when you lean or knock on ball switch.

Lesson 14: 1 digit 7 Segment Displays

Overview

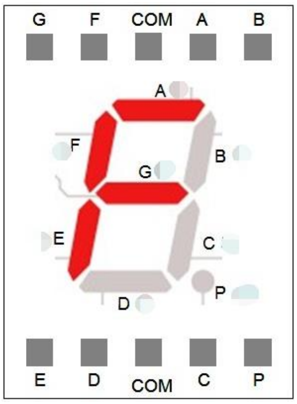


This experiment is similar with the LED experiment, the same is the control of LED, but the experiment can achieve time counting function.

Specification

Null

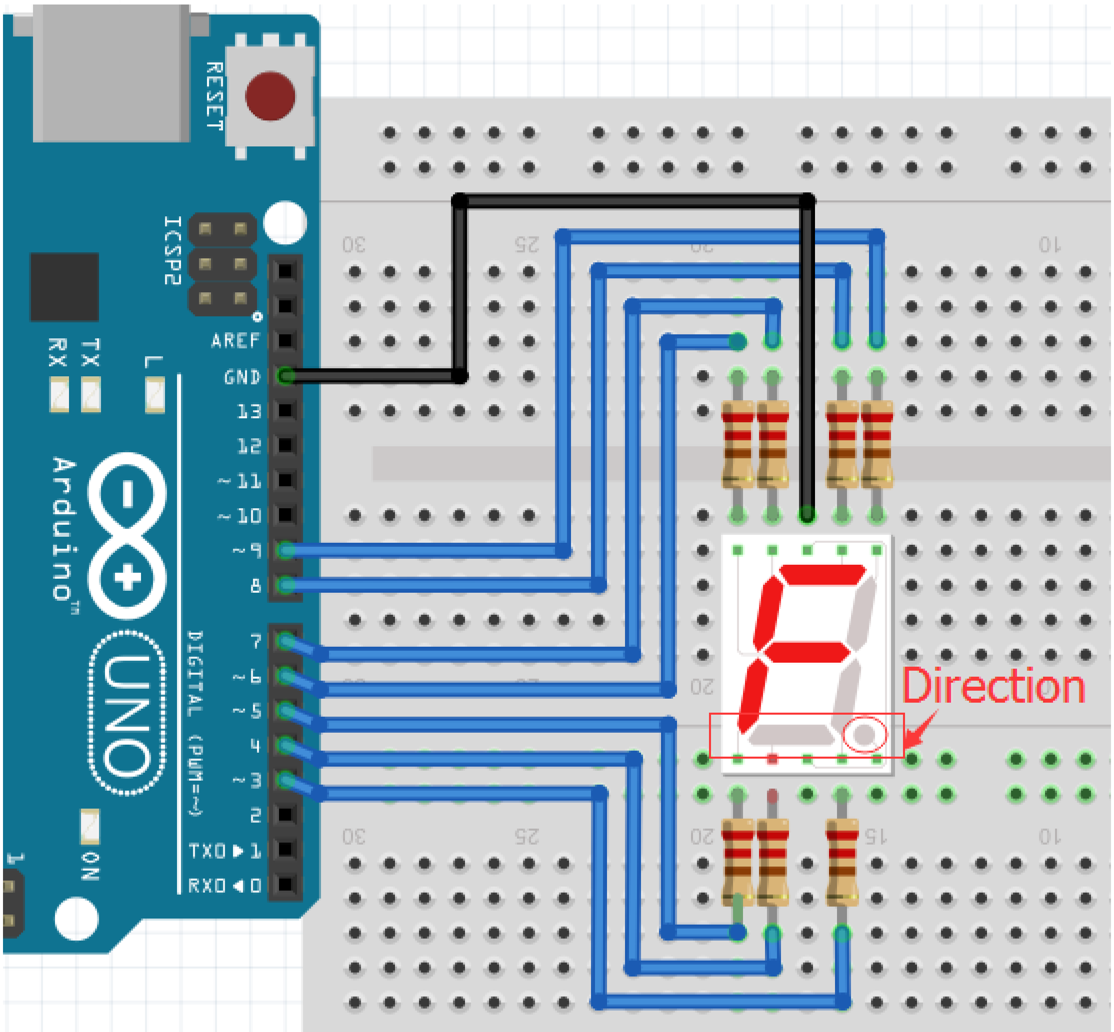
Pin definition



Hardware required

Material diagram	Material name	Number
	1 digit LED Segment Displays	1
	220/330Ω resistor	
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note： Pay attention to the direction of digital tube.

Connection:

UNO R3		SEG
D3	->	C
D4	->	D
D5	->	E
D6	->	G
D7	->	F
D8	->	A
D9	->	B
GND	->	COM

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

[array](#)

Application effect

You will see the number on the digital tube increased from 0 to 9.

Lesson 15: 4 digit 7 Segment Displays

Overview

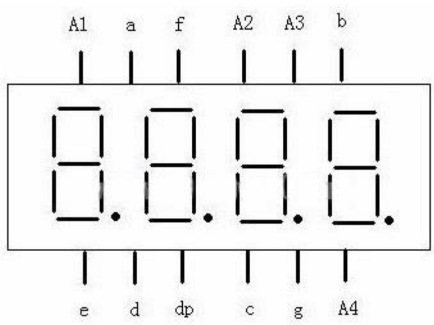


This experiment is similar with the LED experiment, the same is the control of LED, but the experiment can achieve time counting function.

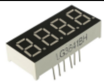
Specification

Null

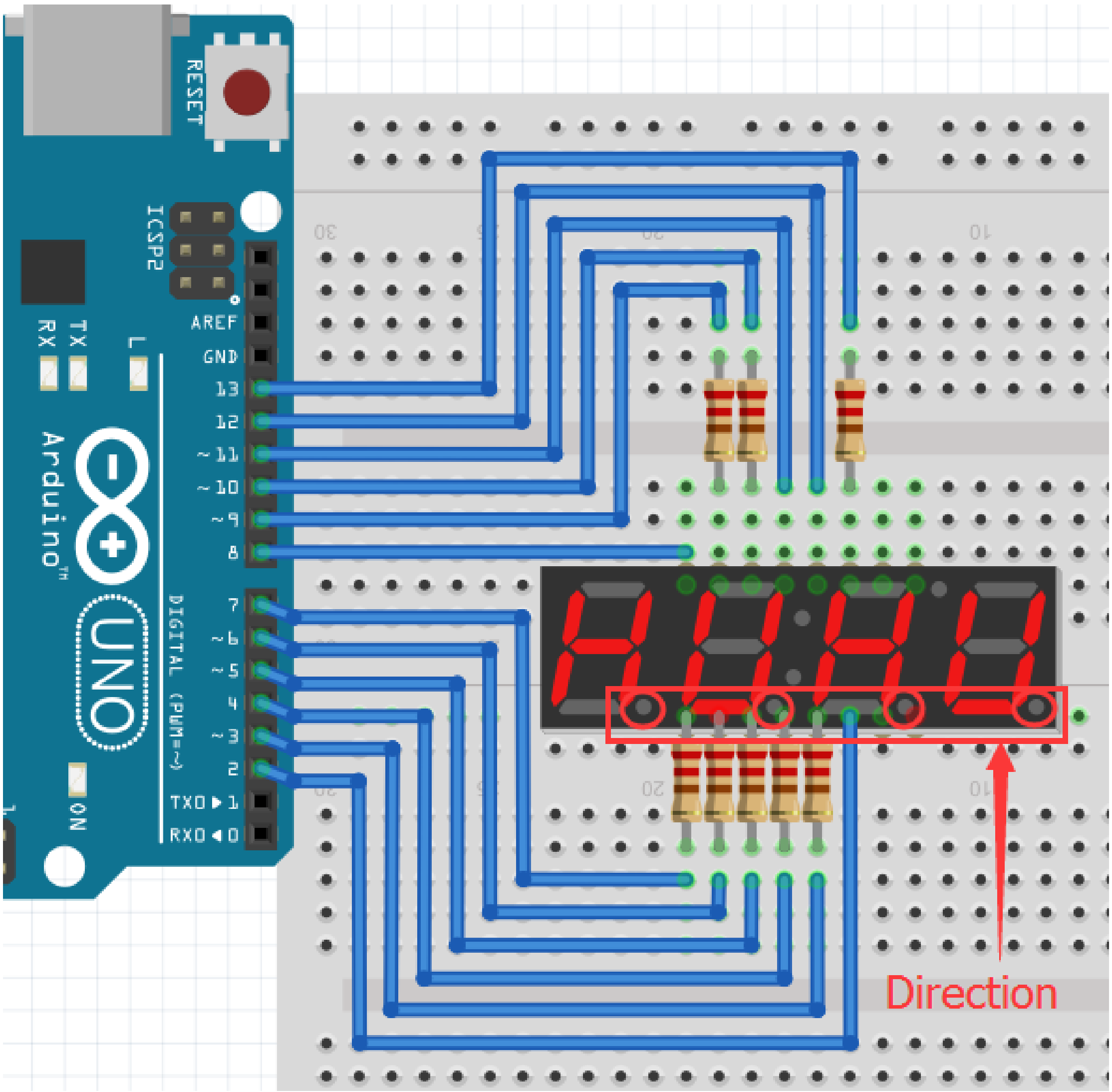
Pin definition



Hardware required

Material diagram	Material name	Number
	4 digit LED Segment Displays	1
	220/330Ω resistor	8
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note： Pay attention to the direction of digital tube.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

[Long](#)
[switch()](#)
[case](#)

Application effect

The time counting function, you will see the number of digital tube display increasingly.

Lesson 16: Heart-shaped display experiment

Overview

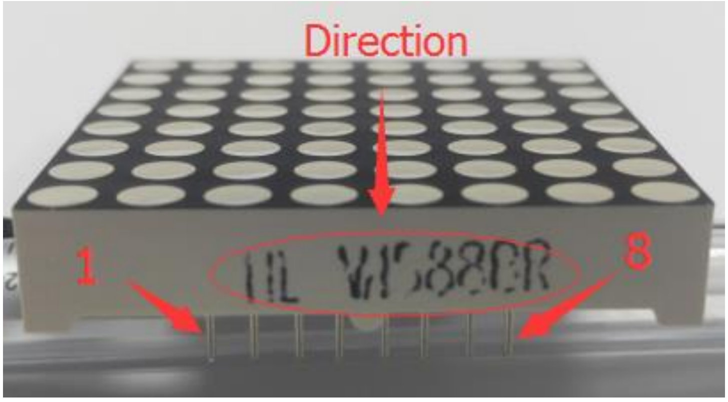


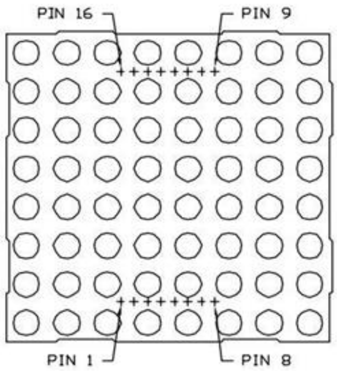
This lesson will teach you how to use an 8*8 dot matrix to display a beating heart animation.

Specification

Please view 1588 ABxx.pdf.
Path: \Datasheet\1588 ABxx.pdf

Pin definition

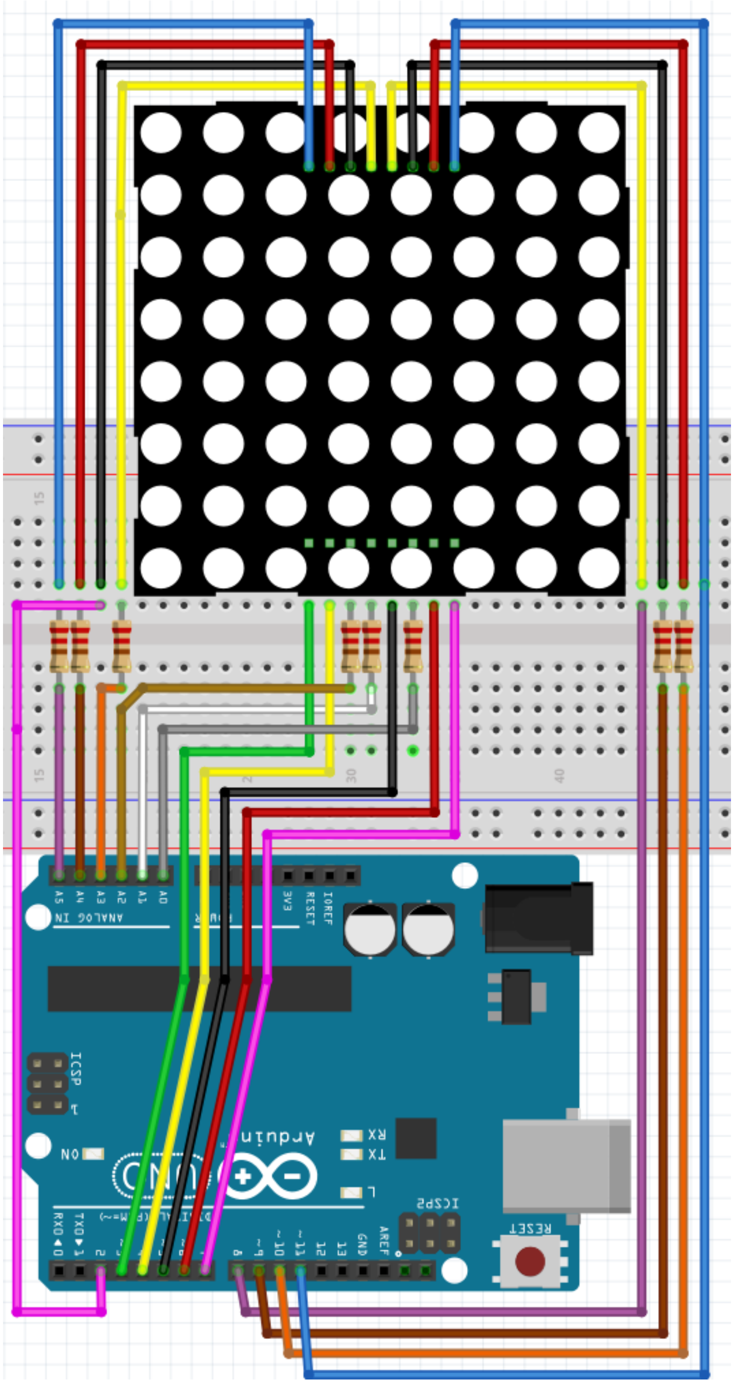




Hardware required

Material diagram	Material name	Number
	8*8 Dot-matrix Display	1
	220/330Ω resistor	8
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Connection:

pin1 ->D3

pin2 ->D4

pin3 ->A2 Need connection resistance.

pin4 ->A1 Need connection resistance.

pin5 ->D5

pin6 ->A0 Need connection resistance.

pin7 ->D6

pin8 ->D7

pin9 ->D11

pin10->D10 Need connection resistance.
pin11- >D9 Need connection resistance.
pin12 ->D8
pin13 ->A3 Need connection resistance
pin14 ->D2
pin15 ->A4 Need connection resistance.
pin16 ->A5 Need connection resistance.

Note： Some pin ports need connection resistance.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
By modifying the "unsigned char table1[8][8] = {}" or "unsigned char table2[8][8] = {}" function, you can display different animation.

Language reference

#define
Unsigned char

Application effect

Please ensure that the connection correct, then upload the code, you will see the heart beating animation.

Lesson 17: Sweep

Overview

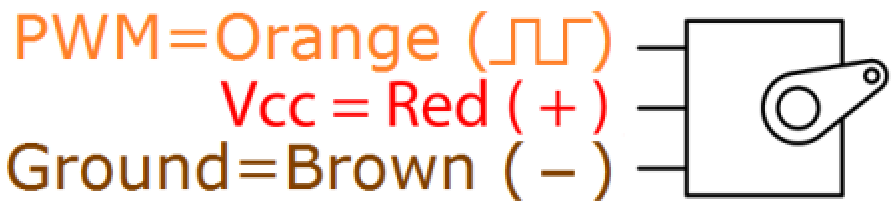


Sweeps the shaft of a RC servo motor back and forth across 180 degrees.
This example makes use of the Arduino servo library.

Specification

Please view SG90Servo-datasheet.pdf.
Path: \ Datasheet\ SG90Servo-datasheet.pdf

Pin definition

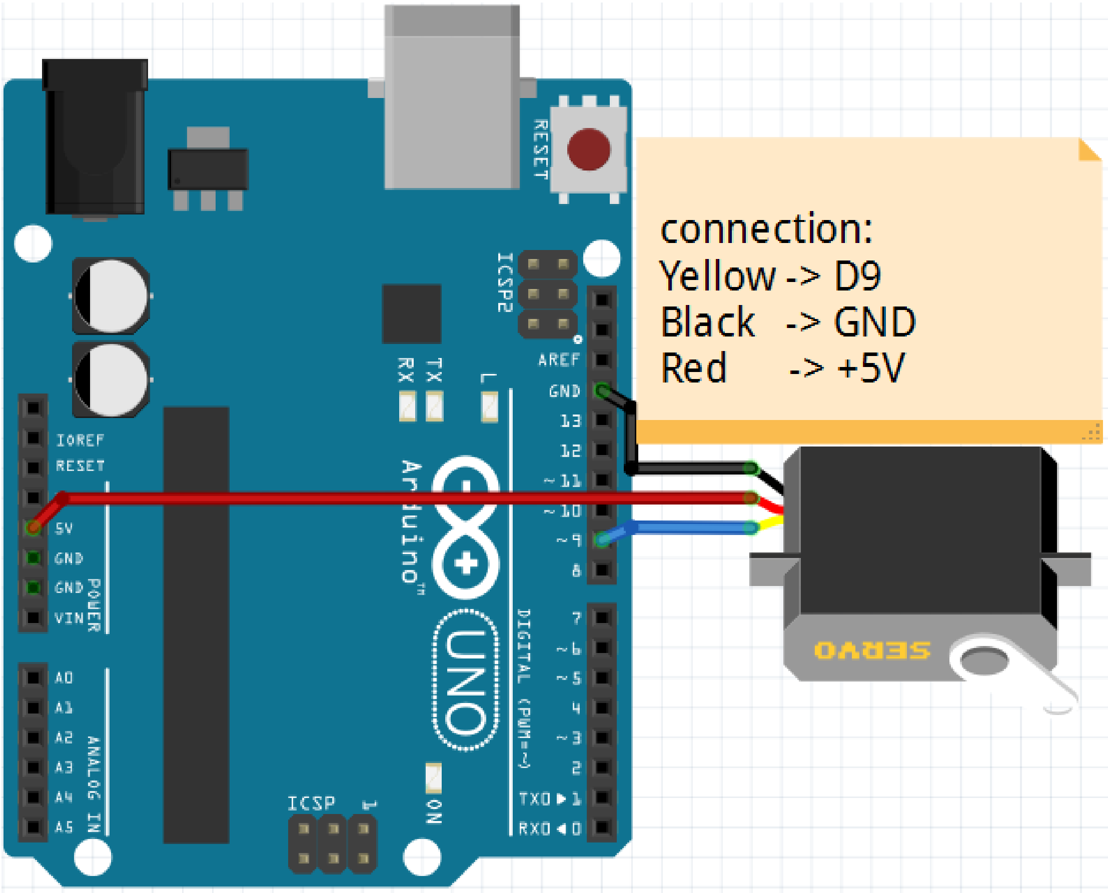


Hardware required

Material diagram	Material name	Number
	9g Servo	1
	USB Cable	1
	UNO R3	1
	Breadboard	1

	Jumper wires	Several

Connection diagram



Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

null

Application effect

You will see the servo motor turning 180 degrees back and forth.

Lesson 18: Knob

Overview



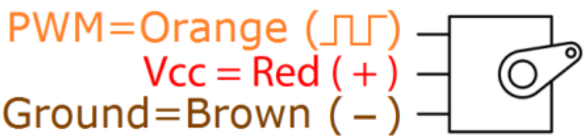
Control the position of a RC (hobby) servo motor with your Arduino and a potentiometer. This example makes use of the Arduino servo library.

Specification

9G servo: please view SG90Servo-datasheet.pdf.
Path: \Datasheet\ SG90Servo-datasheet.pdf
Potentiometer:
Resistance Value: 10K ohm;
Adjustment Type: Top Adjustment

Pin definition

9G servo:



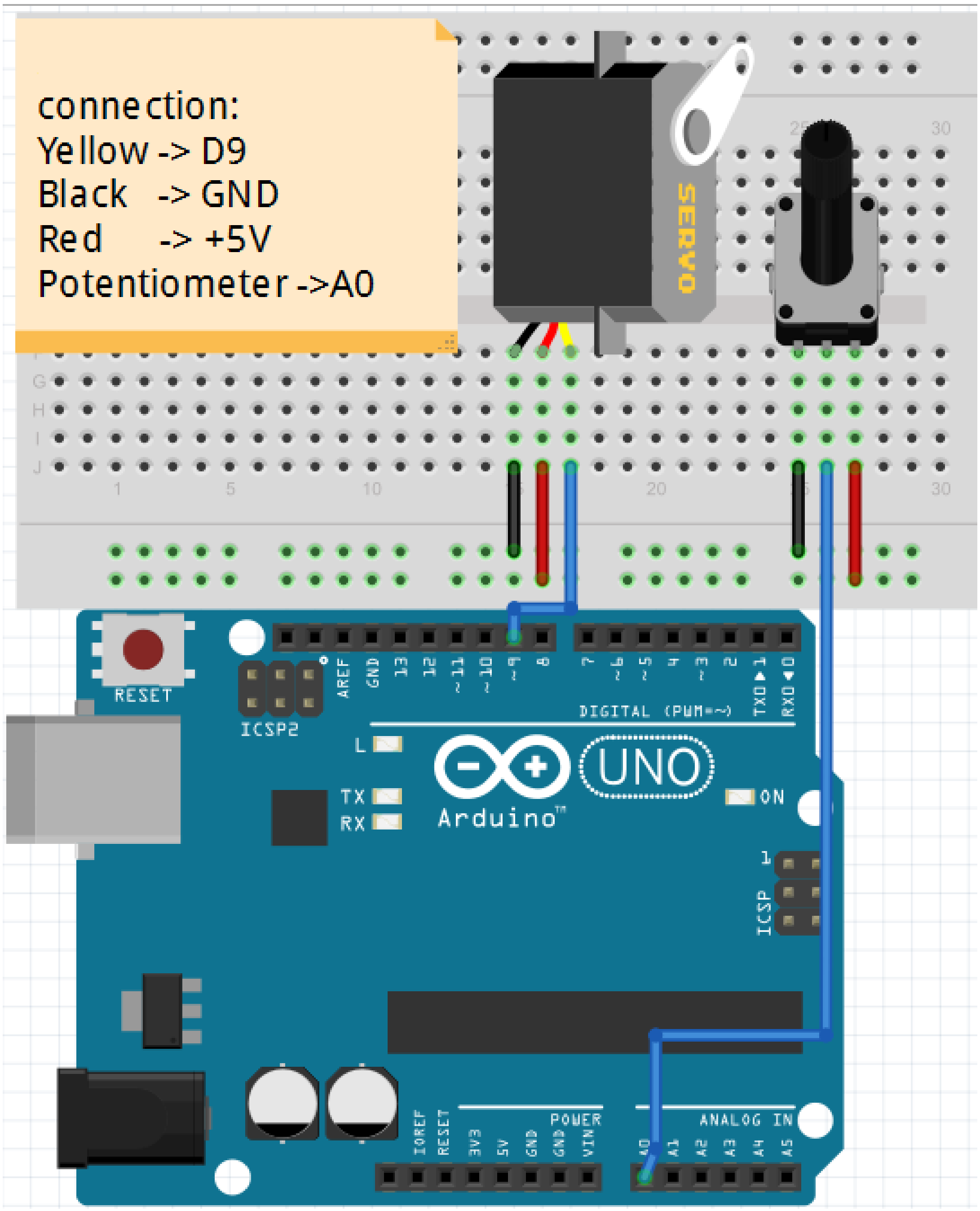
Potentiometer:
Null

Hardware required

Material diagram	Material name	Number
	9g Servo	1
	10KΩ potentiometer	1

	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: The middle pin of the potentiometer is connected to the analog port 0(A0).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
Map()

Application effect

When the rotary potentiometer, the servo motor also with the rotation.

Lesson 19: One step at a time

Overview

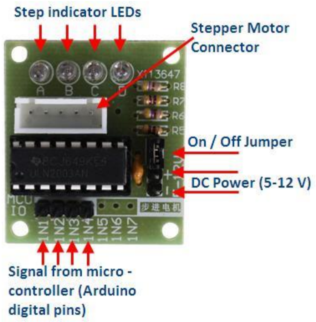


In this lesson, the motor will step one step at a time, very slowly. You can use this to test that you've got the four wires of your stepper wired to the correct pins. If wired correctly, all steps should be in the same direction. You may also use this sketch to count the number of steps that your motor does in one revolution.

Specification

Please view "Stepper-Motor.pdf"
Path: \Datasheet\ Stepper-Motor.pdf

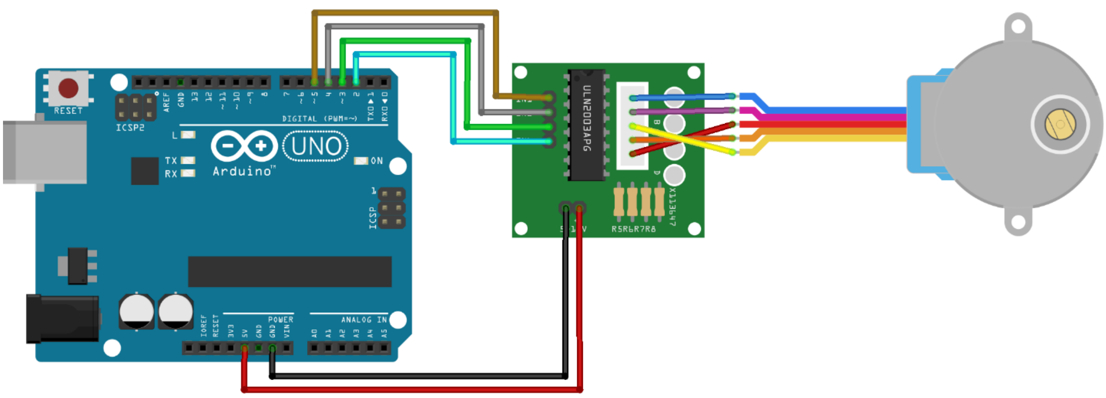
Pin definition



Hardware required

Material diagram	Material name	Number
	Step motor	1
	ULN2003 step motor driver board	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Female to male Jumper wires	6
	Jumper wires	Several

Connection diagram



Connection:

ULN2003 UNO R3

IN4 -> D2

IN3	->	D3
IN2	->	D4
IN1	->	D5
'-'	->	GND
'+'	->	+5V

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Note：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
Stepper myStepper = Stepper(steps, pin1, pin2, pin3, pin4)
stepper.setSpeed()
stepper.step()

Application effect

The motor will step one step at a time, very slowly.

Lesson 20: Stepper speed control

Overview

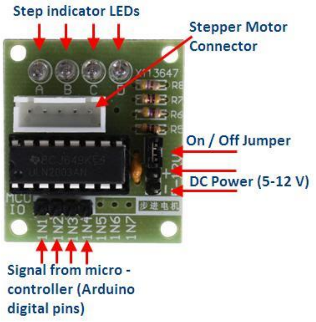


In this lesson, a potentiometer (or other sensor) on analog input 0 is used to control the rotational speed of a stepper motor using the Arduino Stepper Library. The stepper is controlled by with digital pins 2, 3, 4, and 5 for either unipolar or bipolar motors.

Specification

Please view “Stepper-Motor.pdf”
Path: \Datasheet\ Stepper-Motor.pdf

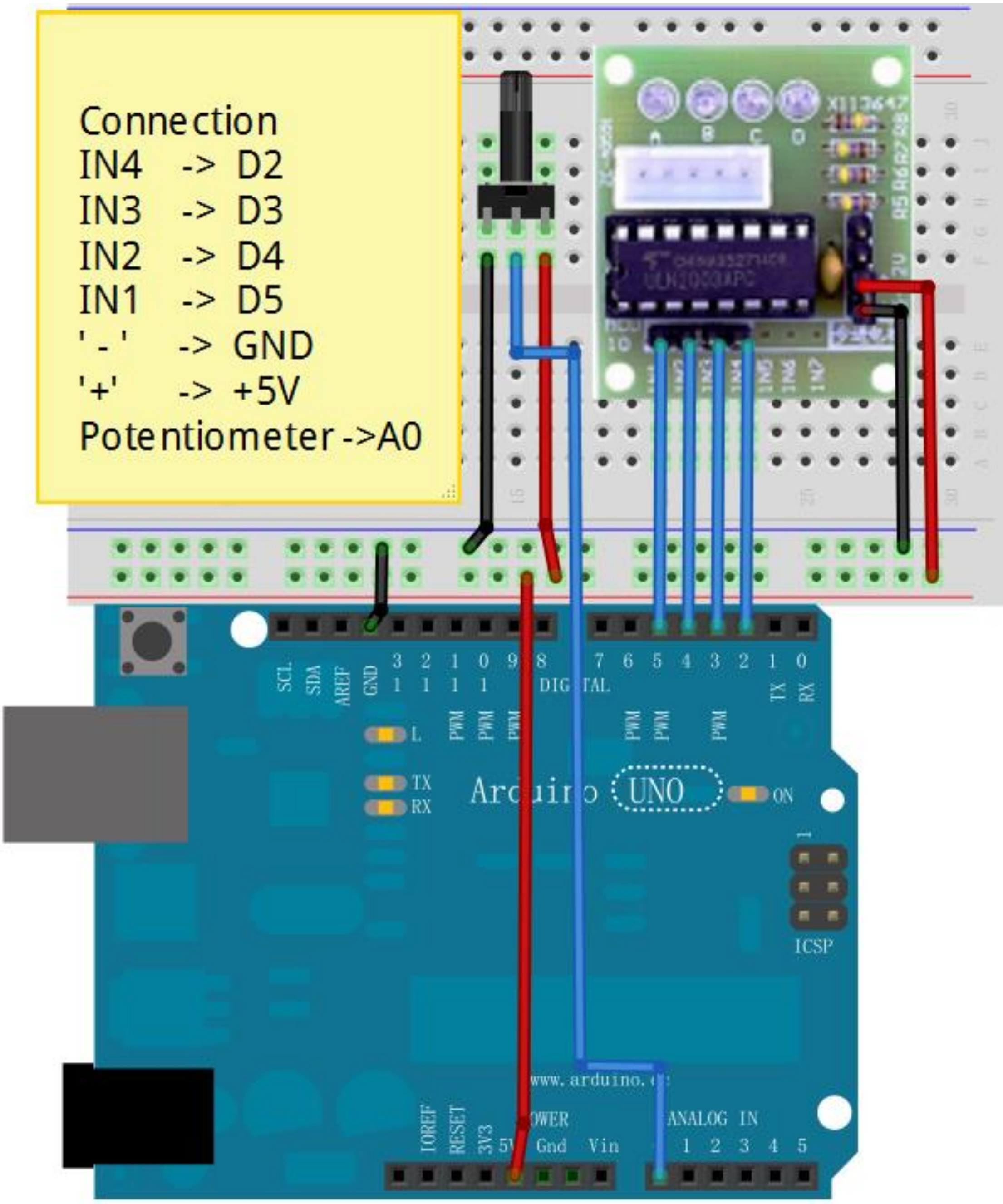
Pin definition



Hardware required

Material diagram	Material name	Number
	Step motor	1
	ULN2003 step motor driver board	1
	10KΩ potentiometer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Female to male Jumper wires	6
	Jumper wires	Several

Connection diagram



Note: The middle pin of the potentiometer is connected to the analog port 0(A0).

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Note：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
Stepper myStepper = Stepper(steps, pin1, pin2, pin3, pin4)
stepper.setSpeed()
stepper.step()

Application effect

The motor will rotate in a clockwise direction. The higher the potentiometer value, the faster the motor speed. Because setSpeed() sets the delay between steps, you may notice the motor is less responsive to changes in the sensor value at low speeds.

Lesson 21: Relay module experiment

Overview



This lesson will teach you how to use a button to control a relay experiment. The Delay() function is not used to eliminate jitter and improve the running efficiency of the program.

Specification

Null

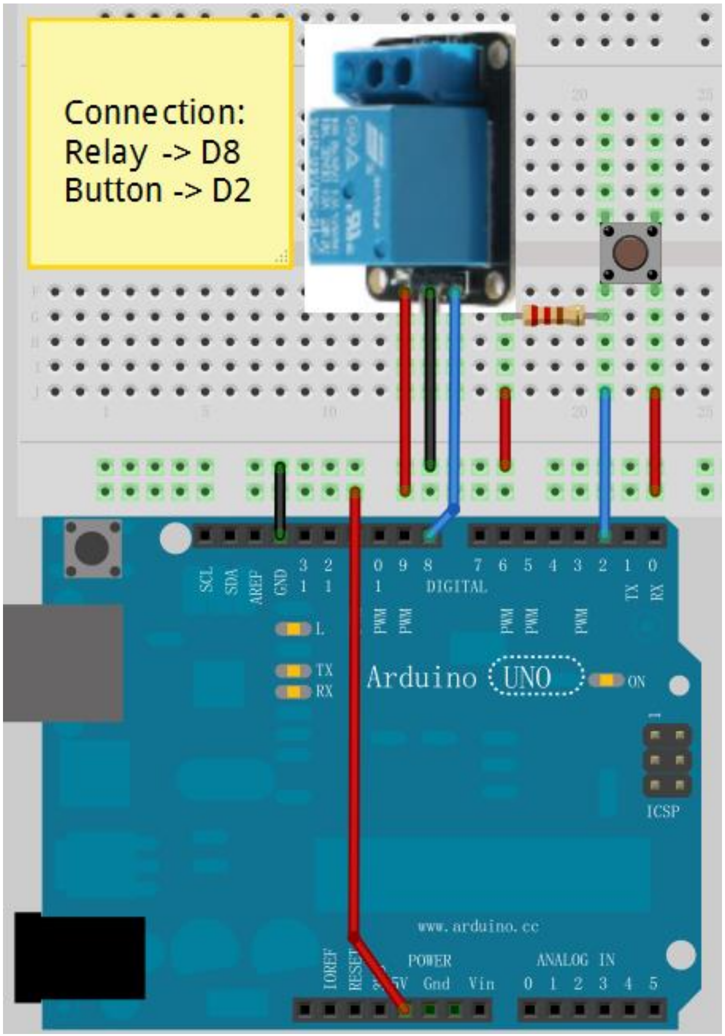
Pin definition

Relay module		UNO R3
S	->	D8
+	->	VCC
-	->	GND

Hardware required

Material diagram	Material name	Number
	Relay module	1
	Button	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

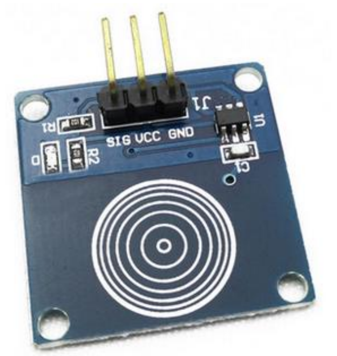
Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
const
millis()

Application effect

When the button is pressed, the state of the relay will be changed.

Lesson 22: Touch lamp

Overview



This is a touch sensor to control the LED lamp experiment, it can control each LED light, but also can achieve the effect of breathing light.

Specification

Null

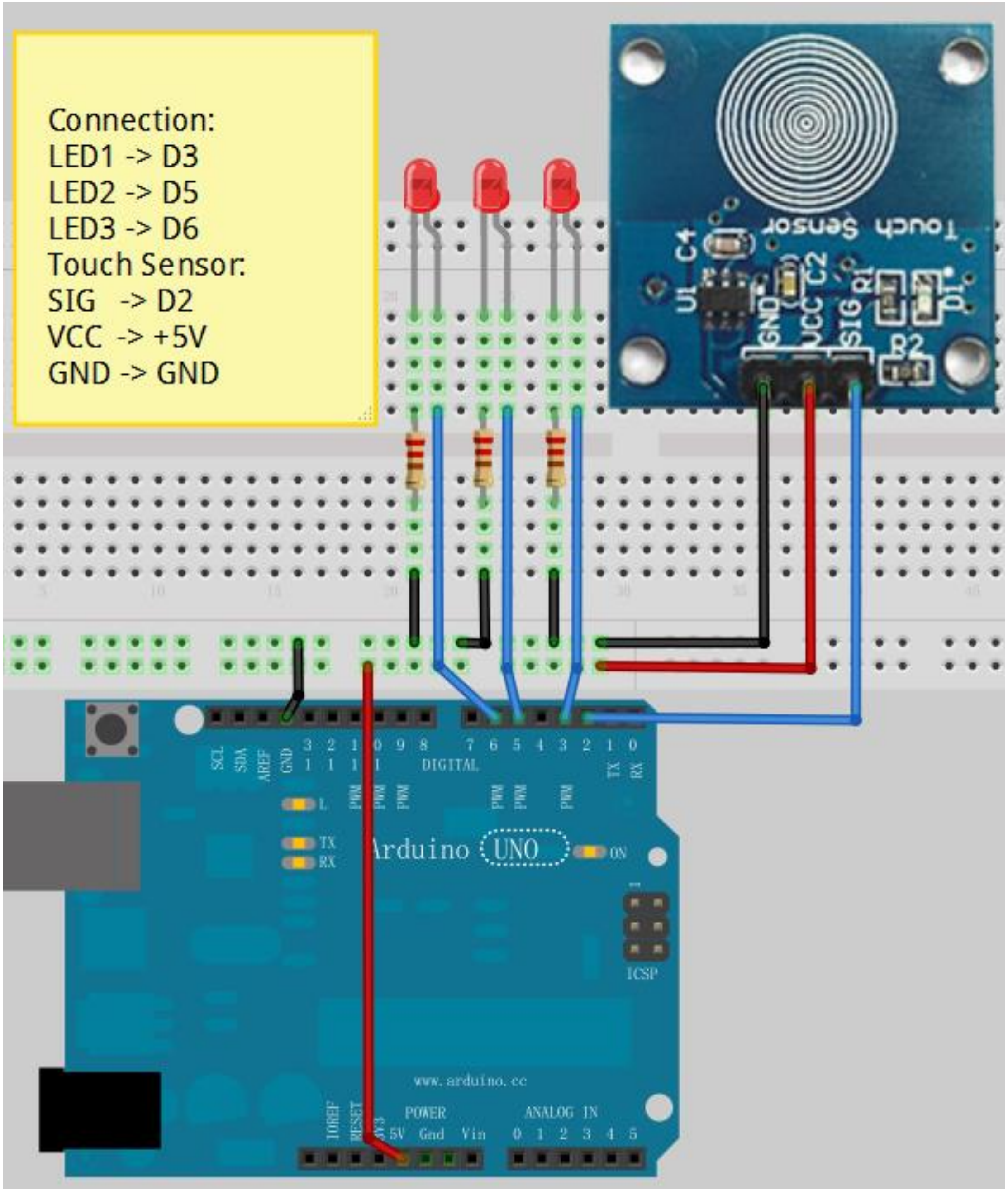
Pin definition

Touch Sensor UNO R3
GND -> GND
VCC -> +5V
SIG -> D2

Hardware required

Material diagram	Material name	Number
	Touch Sensor	1
	LED	3
	220/330Ω resistor	3
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note： The longest LED of the pin is connected to the digital signal port.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
attachInterrupt
switch(case)

Application effect

Through the touch panel, you can control the LED light.

Lesson 23: Flame alarm system

Overview

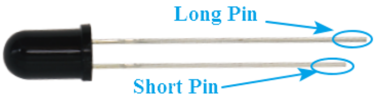


This lesson will teach you how to make a Flame alarm system. It can detect flame.

Specification

Null

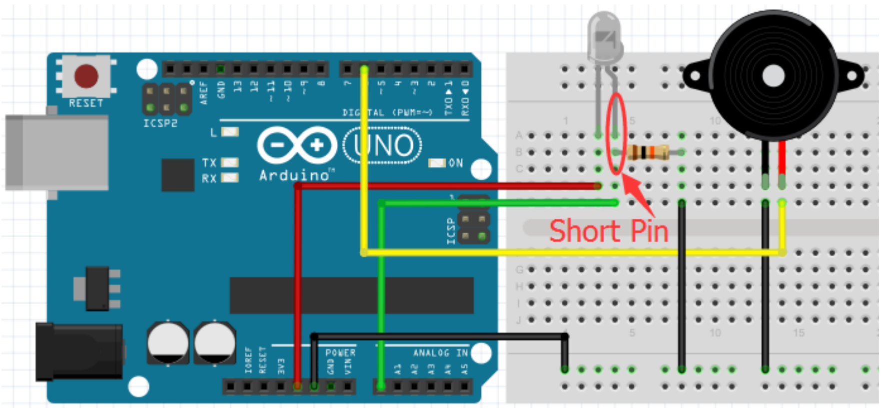
Pin definition



Hardware required

Material diagram	Material name	Number
	Active buzzer	1
	Flame Sensor	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Flame sensor UNO R3

Short Pin -> +5V

Long Pin -> A0

PassiveBuzzer -> D6

Note：The short pin of the Flame sensor is connected to +5v.

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Null

Application effect

We can simulate a flame environment. Turn on the lighter and then near the flame sensor, you will hear the buzzer sound.

Lesson 24: Ultrasonic ranging

Overview



This is the experimental use of ultrasonic module (HCSR04) test distance. Ultrasonic module is generally used in the robot.

Specification

Please view “HCSR04.pdf”
Path: \Datasheet\ HCSR04.pdf

Pin definition

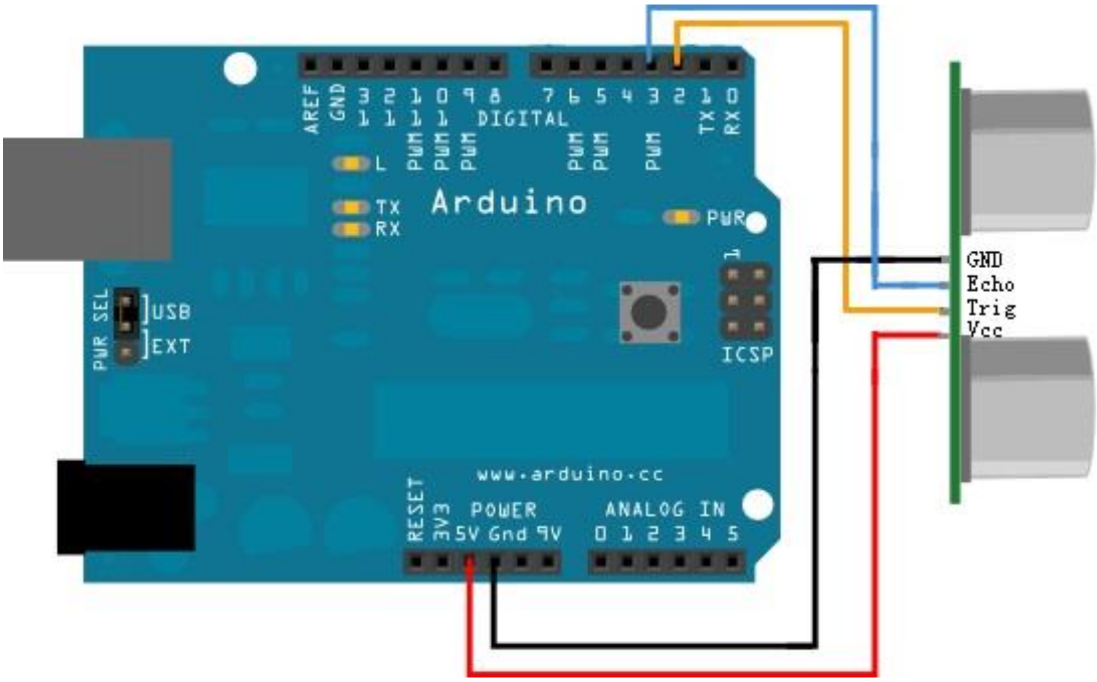
HC SR04		UNO R3
Vcc	->	VCC
Trig	->	D2
Echo	->	D3
Gnd	->	GND

Hardware required

Material diagram	Material name	Number
	HCSR04	1
	USB Cable	1
	UNO R3	1
	Breadboard	1

	Jumper wires	Several

Connection diagram



Connection:

HC SR04		UNO R3
Vcc	->	VCC
Trig	->	D2
Echo	->	D3
Gnd	->	GND

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
[delayMicroseconds()]

Application effect

Open the serial port monitor, and you will see the data returned by the ultrasonic module.

Lesson 25: IR remote control experiment

Overview



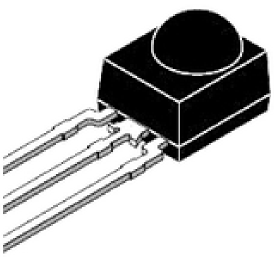


This is an experiment on the infrared reception. This experiment uses the infrared decoder, which involves the content of complex, so only the introduction of the use of methods.

Specification

IR Receiver:
Please view “IR Receiver-datasheet.pdf”
Path: \Datasheet\ IR Receiver-datasheet.pdf

Pin definition



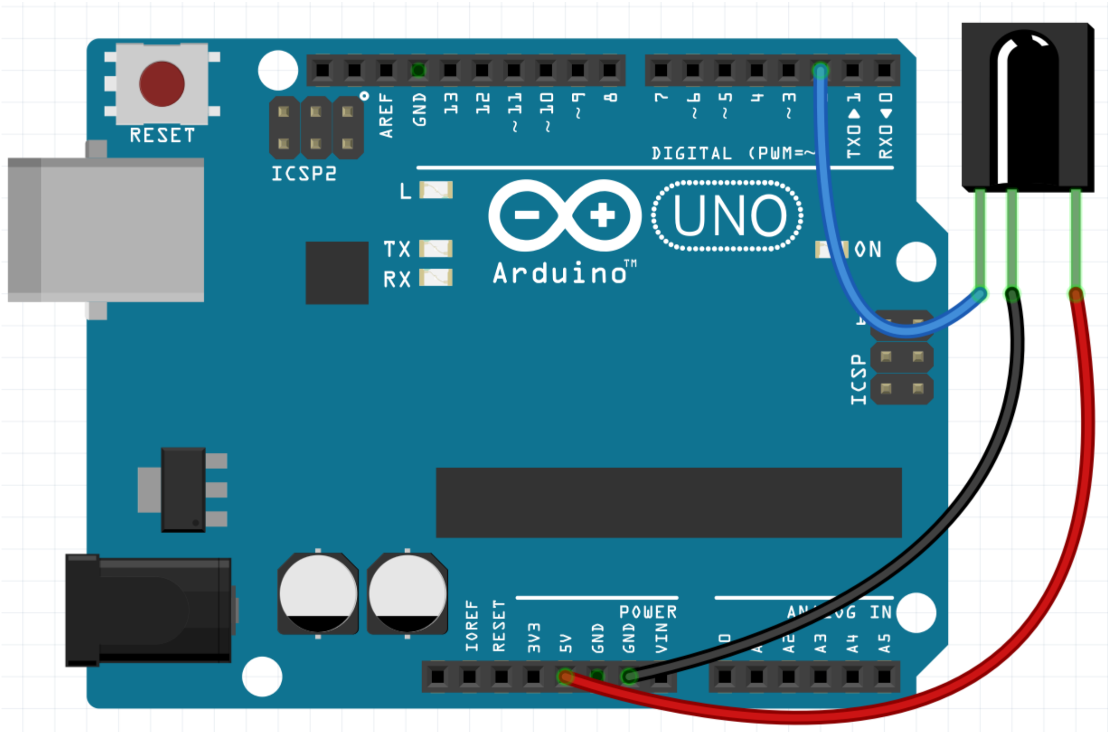


Hardware required

Material diagram	Material name	Number

	IR Remote	1
	IR Receiver	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Note: Please view **Pin definition.**

Connection

UNO R3		IR Receiver
D2	->	OUT
GND	->	GND
+5V	->	VCC

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

If you have added the library, skip it.

Otherwise, you need to add the IRremote to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files'.**

Language reference

Null

Application effect

Open the serial port monitor, press the button of the remote control, you will see that each button will have the corresponding coding.

Lesson 26: LCD1602 with IIC

Overview



This lesson will teach you how to use LCD1602 with IIC.

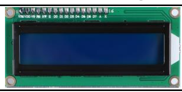
Specification

Please view LCD1602-datasheet.pdf and PCF8574.pdf.
Path: \Datasheet\

Pin definition

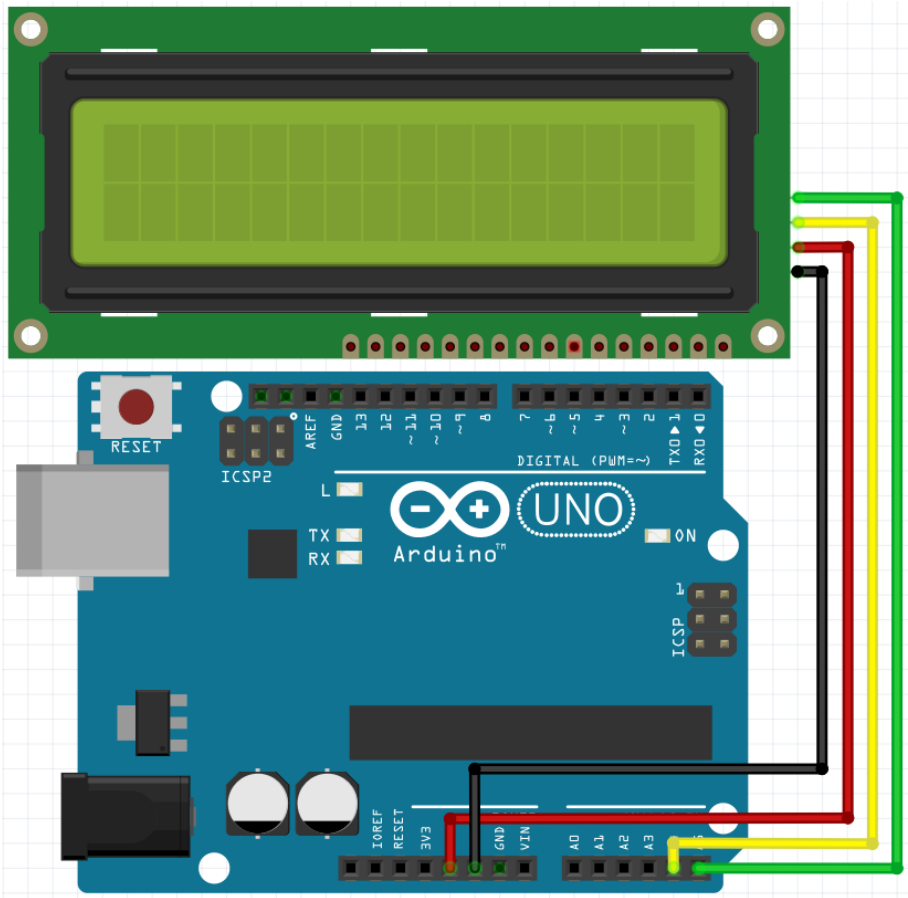
Null.

Hardware required

Material diagram	Material name	Number
	LCD1602_IIC	1
	USB Cable	1
	UNO R3	1

	Breadboard	1
	Jumper wires	Several

Connection diagram



UNO R3		LCD1602_IIC
GND	->	GND
+5V	->	VCC
SDA	->	A4
SCL	->	A5

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
If you have added the library, skip it.
Otherwise, you need to add the **LiquidCrystal_I2C** to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files'.**
If the LCD does not display or brightness is not enough, please adjusted the potentiometer.



Language reference

lcd.begin()
lcd.print()
lcd.setCursor()

Application effect

You will see the LCD display string, while the LCD backlight every 500ms lit once.

Lesson 27: Joystick test experiment

Overview



This lesson will teach you how to use the joystick of the analog output and digital output.

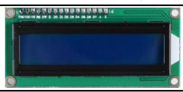
Specification

Two analog outputs, digital outputs.

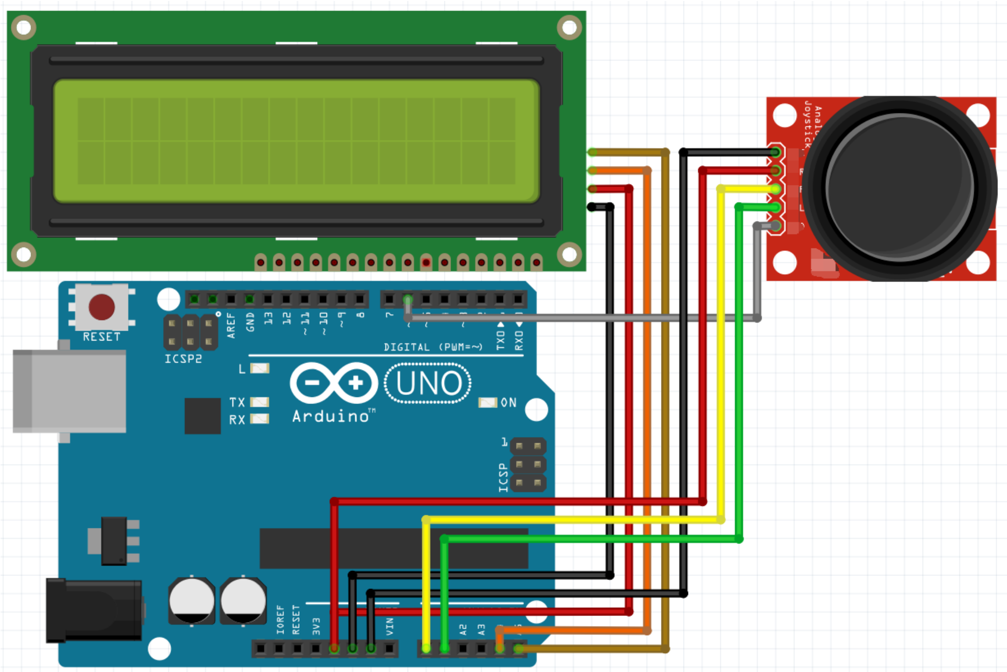
Pin definition

GND -> GND
+5V -> VCC
VRx -> I/O
VRy -> I/O
SW -> I/O

Hardware required

Material diagram	Material name	Number
	LCD1602	1
	Joystick Module	1
	220/330Ω resistor	1
	10KΩ Potentiometer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



LCD1602 with IIC

SCL	->	A5
SDA	->	A4
VCC	->	5V
GND	->	GND

Joystick

GND	->	GND
+5V	->	5V
VRx	->	A0
VRy	->	A1
SW	->	D6

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
If you have added the library, skip it.
Otherwise, you need to add the **LiquidCrystal_I2C** to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files'.**
If the LCD does not display or brightness is not enough, please adjusted the potentiometer.



Language reference

Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
analogRead()

Application effect

By rotating or pressing the joystick, you will see the change in value.
X, Y-axis output of two potentiometers can be read through the AD converter twist angle.
Press down on the joystick, touch switches can be deployed all the way, as a digital output, has a pull-up.

Lesson 28: Water level monitoring experiment

Overview



This is a water level measurement experiment, it is relatively simple to achieve, only need to read the value of the analog port(A0), and then converted to a percentage.

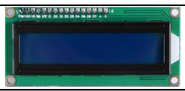
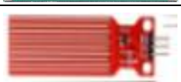
Specification

Operating voltage: DC3-5V
Operating current: less than 20mA
Sensor Type: Analog
Production process: FR4 double-sided HASL
Humidity: 10% -90% non-condensing
Detection Area: 40mmx16mm
Product Dimensions: 62mmx20mmx8mm

Pin definition

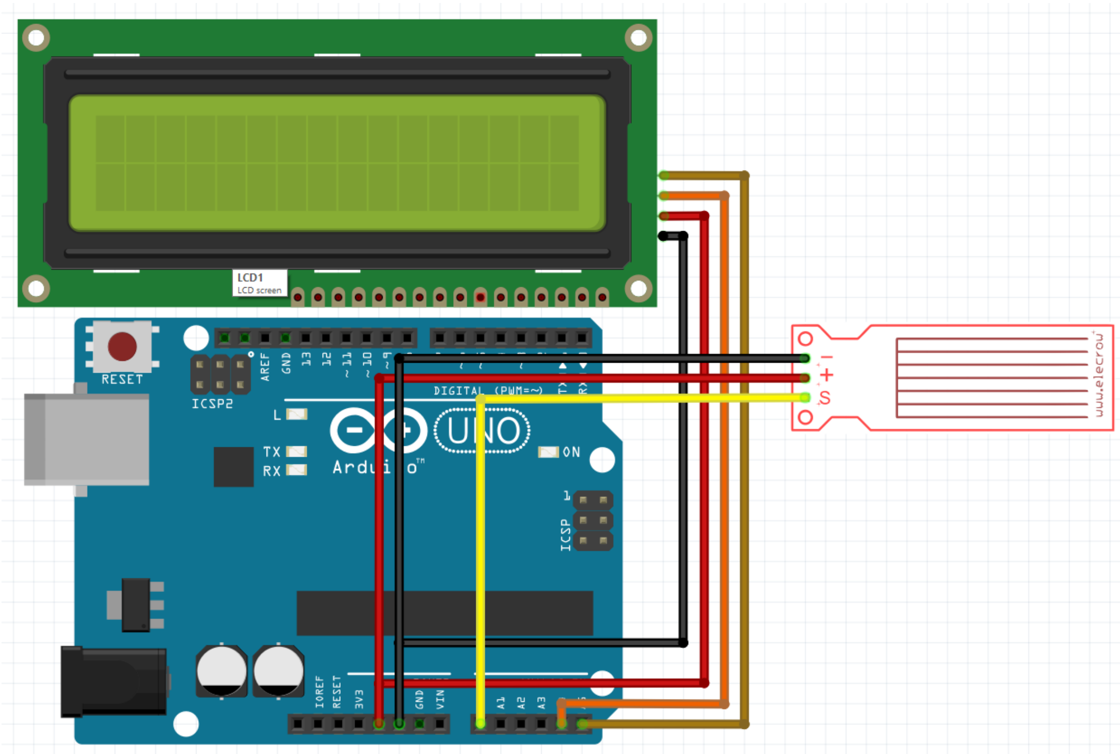
Null.

Hardware required

Material diagram	Material name	Number
	LCD1602	1
	Water Lever Sensor	1

	220/330Ω resistor	1
	10KΩ Potentiometer	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



UNO R3 Watersensor

GND -> -

5V -> +

A0 -> S

UNO R3 LCD1602_IIC

GND -> GND

+5V -> VCC

SDA -> A4

SCL -> A5

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).

If you have added the library, skip it.
Otherwise, you need to add the **LiquidCrystal_I2C** to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files.docx'.**
If the LCD does not display or brightness is not enough, please adjusted the potentiometer.



Language reference

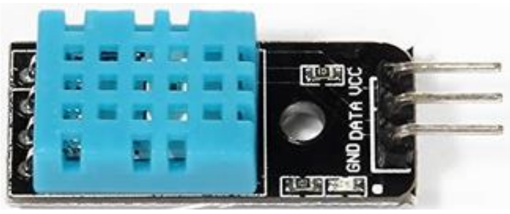
Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
analogRead()

Application effect

When the water level sensor enter water and make it of different height, the LCD will display different percentage.

Lesson 29: DHT11 experiment

Overview



This is an experiment on temperature and humidity, you will learn the external library files to simplify the process.

Specification

Please view DHT11-datasheet.pdf.
Path: \Datasheet\ DHT11-datasheet.pdf

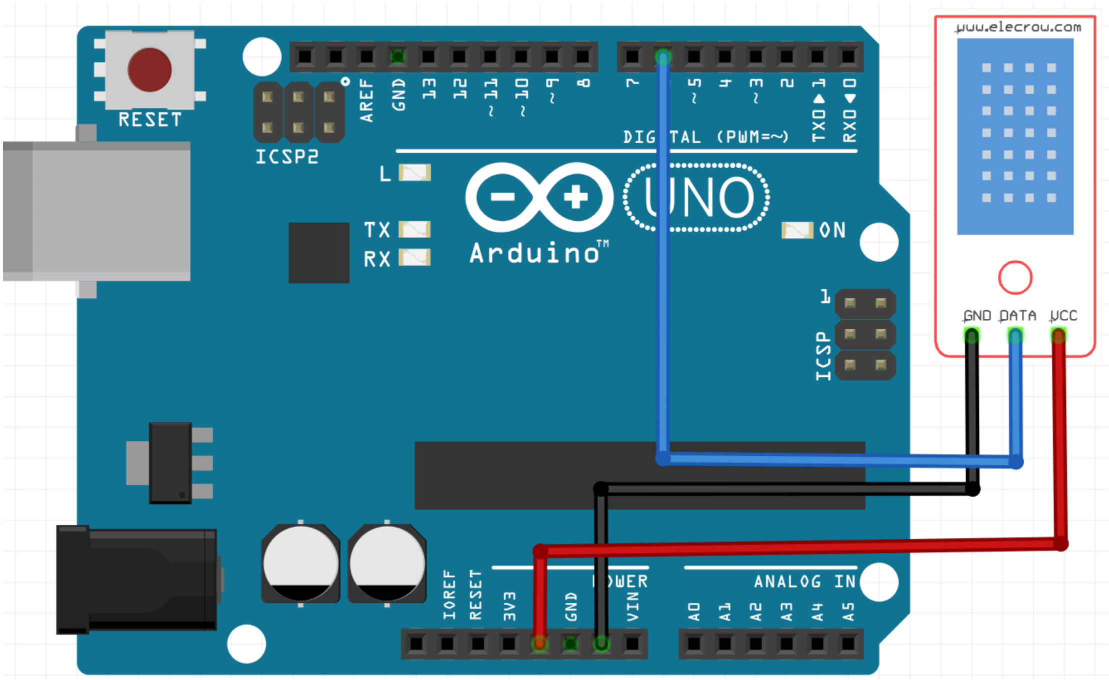
Pin definition

UNO R3		DHT11
GND	->	GND/'-'
D6	->	DATA/'out'
+5V	->	VCC/'+'

Hardware required

Material diagram	Material name	Number
	DHT11	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
If you have added the library, skip it.

Otherwise, you need to add the **DHT11** to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files.docx'.**

Language reference

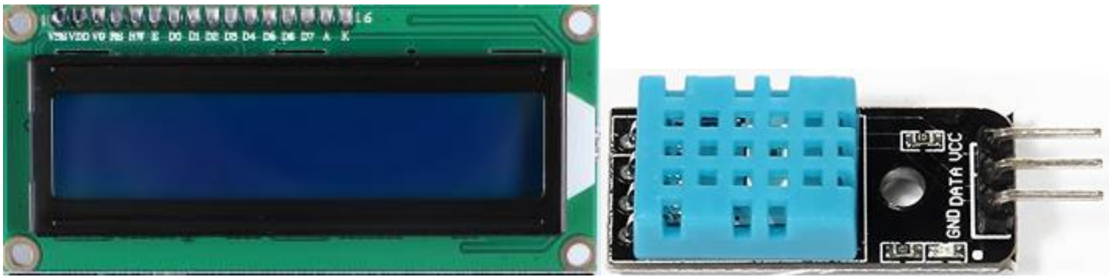
Tips：click on the following name to jump to the web page.
If you fail to open, use the Adobe reader to open this document.
serial

Application effect

Open the serial port monitor, you will see the value returned by DHT11.

Lesson 30: Temperature and humidity monitoring experiment

Overview



This is a more complex experiment, it can realize the monitoring of indoor temperature and humidity, and in the LCD above display value.

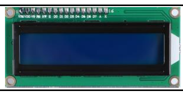
Specification

Please view LCD1602-datasheet.pdf、DHT11-datasheet.pdf and PCF8574.pdf.
Path: \Datasheet

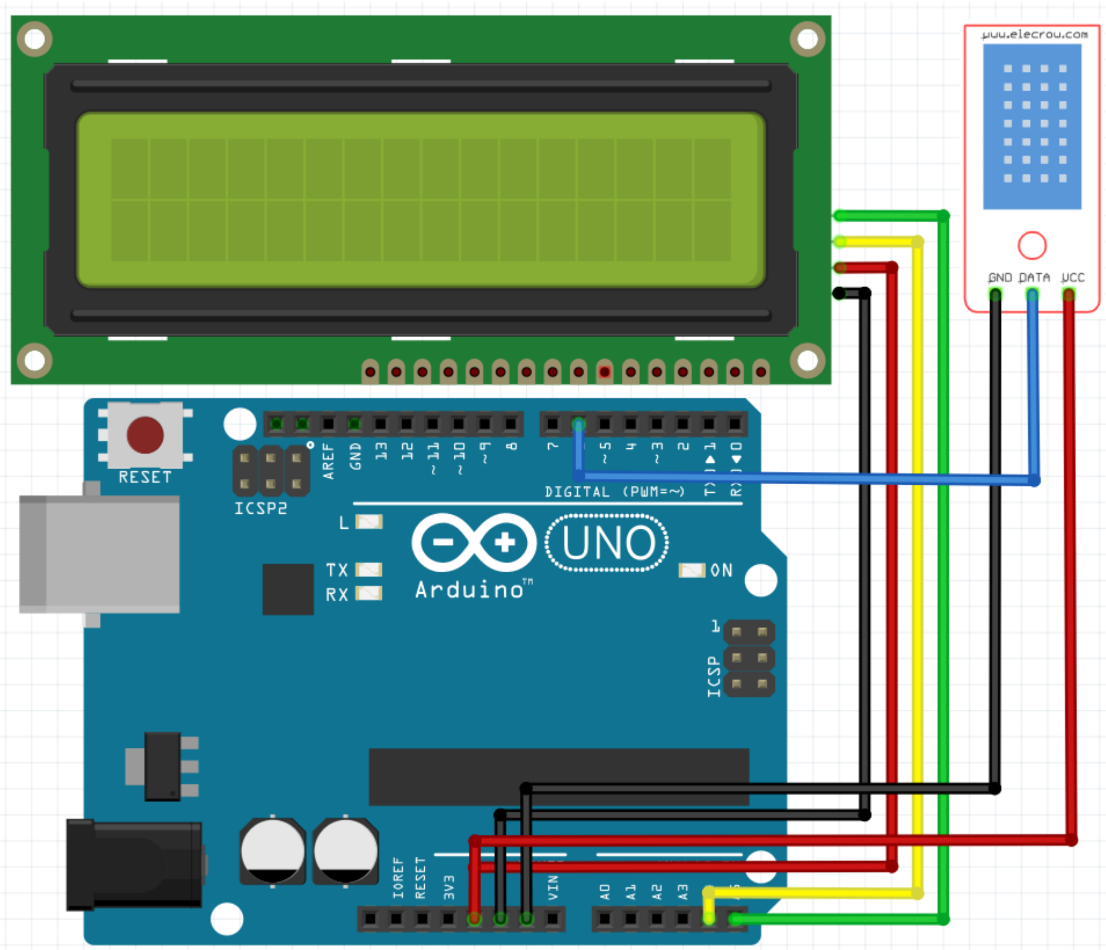
Pin definition

Null

Hardware required

Material diagram	Material name	Number
	LCD1602_IIC	1
	DHT11	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

Connection diagram



UNO R3		DHT11
GND	->	GND/'-'
D6	->	DATA/'out'
+5V	->	VCC/'+'

UNO R3		LCD1602_IIC
GND	->	GND
+5V	->	VCC
SDA	->	A4
SCL	->	A5

Compile and upload

Tips: Refer to the operation demo (Step4 to Step8).
If you have added the library, skip it.
Otherwise, you need to add the **LiquidCrystal_I2C and DHT11** to the Arduino library file directory, otherwise the compiler does not pass. **Please refer to 'How to add library files.docx'.**
If the LCD does not display or brightness is not enough, please adjusted the potentiometer.



Language reference

lcd.begin()
lcd.print()
lcd.setCursor()

Application effect

You will see the value of temperature and humidity on the LCD.